commodore Magazine

AÑO I - Núm. 12 -
Febrero 1985 - 250 Ptas.

REVISTA INDEPENDIENTE PARA USUARIOS

2a parte
Programa-aventura:
SUPERVIVENCIA

Commodore-16: por dentro y por fuera

SPRITES: Los alegres duendecillos

commodore *Magazine*

Sumario

Director:
Alejandro Diges
Coordinador editorial:
J. A. Sanz
Redacción:
Roberto Menéndez
Aníbal Pardo
Teresa Aranda
Gumersindo García
Fernando García
Diseño:
Ricardo Segura

Editada por
Publinformática
Presidente:
Fernando Bolín
Director Editorial:
Norberto Gallego

Administración:
INFODIS S.A.
Gerente de Circulación y Ventas:
Luis Carrero
Producción:
Miguel Onieva
Director de Marketing:
Antonio González
Servicio al cliente:
Julia González - Tel.: 733 79 69
Administración:
Miguel Atance, Antonio Torres
Jefe de Publicidad:
María José Martín

Dirección y Redacción
C/Bravo Murillo, 377 - 5º A
Tel.: 733 74 13
28020 - MADRID
Publicidad y Administración:
C/ Bravo Murillo, 377 - 3º E
Tel. 733 96 62/96

Publicidad en Madrid:
Fernando Hernando
Publicidad en Barcelona:
María del Carmen Ríos
y Jorge González
Pelayo, 12.
Tel.: (93) 301 47 00 ext. 27 y 28
08001 - BARCELONA
Depósito Legal: M-6622-1984
Distribuye: S.G.E.L.
Avda. Valdelaparra, s/n.
Alcobendas. Madrid

Fotocomposición: Consulgraf
C/Nicolás Morales, 34. Madrid
Fotomecánica: Karmat
C/Pantoja, 10. Madrid
Imprime: Novograph, S.A.
Carretera de Irún, Km. 12,450
Madrid.
Solicitado control O.J.D.

Esta publicación es miembro de la Asociación de Revistas de la Información ARI, asociada a la Federación Internacional de Prensa Periódica FIPP.

Año 1
Núm.12

ROGAMOS DIRIJAN TODA LA CORRESPONDENCIA RELACIONADA CON SUSCRIPCIONES A:
COMMODORE MAGAZINE
EDISA: Tel. 415 97 12
C/ López de Hoyos, 141-5.
28002-MADRID
PARA TODOS LOS PAGOS RESEÑAR SOLAMENTE COMMODORE MAGAZINE

ROGAMOS QUE PARA LA COMPRA DE EJEMPLARES ATRASADOS SE DIRIJAN A LA PROPIA EDITORIAL
commodore Magazine
C/ Bravo Murillo, 377-5. A
Tel. 733 74 13
28020-MADRID

ari

Esta revista no mantiene relación de dependencia de ningún tipo con respecto de los fabricantes de ordenadores Commodore Business Machines ni de sus representantes.

Editorial

Las fechas previas a las pasadas Navidades sufrimos un verdadero aluvión de cartas con idéntico fin. No era otro que preguntarnos sobre esas ofertas que han aparecido por doquier de ordenadores con precio por debajo del anunciado por el distribuidor en nuestro país. La variedad de precios y gangas resulta de lo más variada. **Commodore Magazine** desconoce a ciencia cierta cuál es la procedencia de tales máquinas, pero es de suponer que de los canales alternativos de distribución. Esto es algo que a nosotros no nos compete, por estar totalmente desvinculados del fabricante y sus representantes (como reza la leyenda de nuestra página). Sin embargo, bien podemos dar un consejo: «No hay quien dé duros por pesetas, como dice un antiguo y conocido refrán. Tal vez estamos ahorrando el precio de la garantía y el servicio postventa. Es conveniente indagar sobre lo que ocurrirá en el futuro si nuestro flamante ordenador se avería. Nadie mejor que el vendedor para hablarnos de la garantía.

También nos escribís interesándoos por los programas y cartas con preguntas que habéis enviado. Lo cierto es que al cabo del día se reciben bastantes y es por tanto preciso guardar un orden para su publicación.

Otras veces, como ocurre con las cartas, se reúnen varias cuyo contenido guarda una cierta relación entre sí, contestando a la más representativa en nombre de todas.

La calidad de los programas que nos hacéis llegar es cada vez mayor, pero esto no quiere decir que los publiquemos inmediatamente; también habrán de esperar su turno. Todo esto es aplicable a las simpáticas fotos de vuestro rincón de trabajo.

El recién llegado, el tan esperado **C-16**, es el tema principal de este número de la revista. Por lo que hemos visto puede ser un magnífico ordenador para adentrarnos en el mundo microinformático de la mano de un ordenador versátil y bonachón. Después de leer el artículo pretendemos que saquéis vuestras propias conclusiones.

Los *Sprites* para el **C-64** son otro de los temas de continua preocupación para quienes desarrolláis vuestros propios programas con gráficos en movimiento. Un nuevo artículo en torno a los pequeños duendes electrónicos es lo que ofrecemos unas cuantas páginas más adelante.



Códigos de control para el VIC-20 y el C-64

COMO SE VE	COMO SE TECLEA	EFECTO CONSEGUIDO
COLORES DEL VIC-20 Y DEL 64		
	CTRL+1	NEGRO
	CTRL+2	BLANCO
	CTRL+3	ROJO
	CTRL+4	CIAN
	CTRL+5	PURPURA
	CTRL+6	VERDE
	CTRL+7	AZUL
	CTRL+8	AMARILLO

COLORES DEL 64 SOLAMENTE

	CBM+1	NARANJA
	CBM+2	MARRON
	CBM+3	ROSA
	CBM+4	GRIS OSCURO
	CBM+5	GRIS MEDIO
	CBM+6	VERDE CLARO
	CBM+7	AZUL CLARO
	CBM+8	GRIS CLARO

CODIGOS DE CURSOR Y CONTROL

	HOME	CURSOR A CASA
	SHIFT+HOME	LIMPIA PANTALLA
	CRSR	CURSOR DERECHA
	SHIFT+CRSR	CURSOR IZQUIERDA
	CRSR	CURSOR ABAJO
	SHIFT+CRSR	CURSOR ARRIBA
	CTRL+9	CARACTER INVERSO
	CTRL+0	CARACTER NORMAL
	DEL	BORRAR
	SHIFT+DEL	INSERTAR

TECLAS DE FUNCION

	F1	
	F2=SHIFT+F1	
	F3	
	F4=SHIFT+F3	
	F5	
	F6=SHIFT+F5	
	F7	
	F8=SHIFT+F7	

Cartas

P.: Soy un lector de vuestra y «nuestra» revista que os escribe desde la provincia de Barcelona. Bueno, el caso es que yo me compro muchas revistas de informática para el **C-64**, y he encontrado una cosa a faltar (aunque os parezca una tontería), que ni siquiera figura en el manual: ¿Cuánto tiempo se puede tener encendido el ordenador? porque, si no se dice, lo mismo se puede tener encendido una hora que diez. También os quería comentar otra cosa: ¿por qué no poneías a «todos» los programas de concurso unas líneas para que puedan manejarse desde el teclado sin tener que utilizar necesariamente un *joystick?*

C. Valdegate
Barcelona

R.: Lo que nos planteas es una interesante pregunta, cuya respuesta no todo el mundo tiene demasiado clara. En principio, no hay ningún límite sobre el tiempo que un ordenador puede estar encendido de forma contínua; este tiempo puede ser semanas, meses o incluso años. Los constructores de la máquina ya se han encargado de que el régimen de funcionamiento sea estable, es decir, el ordenador no se va calentando más y más con el tiempo, sino que alcanza una temperatura de funcionamiento que se mantiene estable. Esta temperatura se alcanza ya a los pocos minutos de encender el aparato. El único límite, en principio, es «hasta que falle algún componente». Cada uno de estos tiene su tiempo de vida media, que es un parámetro estadístico sobre el tiempo que se espera que el componente funcione correctamente. Claro que cada componente concreto puede vivir más o menos tiempo que ese valor medio, y, en general, vivirá menos cuanto mayor sea la temperatura a la que esté trabajando. Así que, si el ordenador está apagado, la temperatura de los componentes es menor que cuando está encendido, y su desgaste también es menor. Esto nos lleva a la conclusión de que, en general, cuanto más tiempo esté encendido el ordenador, mejor, más durará.

En resumen, aunque el ordenador puede estar encendido todo el tiempo que se quiera sin que ello signifique que se va a estropear, la política más inteligente es la de apagarlo cuando no sea imprescindible que éste encendido.

Con respecto a la segunda parte de tu pregunta, creemos que nosotros no debemos modificar los programas que nos envían a concursar, sino que estos deben ir publicado tal y como nos los envían sus autores. Lo que sí podemos hacer es animar a todos aquellos que nos envían programas a que sean ellos mismos lo que incluyan rutinas tanto para *joystick* como para teclado.

P.: Les ruego que tengan a bien informarme sobre la diferencia que existe entre un ordenador **C-64** y un **CBM-64**, y si son válidos para ambos los mismos programas. He preguntado en varios establecimientos de esta ciudad y desconocen el modelo **CBM-64**.

M. A. Vázquez
Linares (Jaén)

R.: No existe ninguna diferencia entre **C-64** y **CBM-64**, ambos son abreviaturas de la misma máquina, que es el modelo **Commodore-64** de la compañía **Commodore Business Machines** (de donde vienen las abrevituras **CBM**). Es decir, no son dos máquinas distintas, sino dos nombres para la misma máquina, y por lo tanto todos los programas para **C-64** valen para **CBM-64** (no es que valgan, es que son para **CBM-64**, que es exactamente la misma máquina que el **C-64**).

P.: Mi duda es la siguiente: ¿Cómo se realiza la programación de las teclas de función F1-F8?

L. Mur
Huesca

R.: Las teclas de función, tanto en el **Commodore-64** como en el **Vic-20**, están previstas para que el usuario-programador las utilice como más le convenga. En general, el empleo de estas teclas se lleva a cabo desde dentro de un programa en BASIC, de forma que, al pulsar la tecla F1 por ejemplo, el programa salte a una subrutina determinada que haga tal o cual cosa. Para conseguir ésto, lo más cómodo es emplear un par de instrucciones como las siguientes:

```
GET A$
IF A$ = CHR$(133) THEN GOSUB...
```

La primera línea se encarga de leer el teclado y almacenar en A$ el valor correspondiente a la tecla pulsada. La segunda línea es la que importa ya que indica que si la tecla pulsada es aquella cuyo código CHR$ vale 133 (esta tecla es F1) el programa debe saltar a la subrutina correspondiente. Los códigos CHR$ correspondientes al resto de las teclas de función son los siguientes:

F2	137		
F4	138	F3	134
F6	139	F5	135
F8	140	F7	136

Otra forma de hacer esto mismo es sustituyendo la segunda línea por ésta:

```
IF A$ = " Símbolo que se obtiene al pulsar F1 en modo comillas" THEN GO SUB...
```

Es decir, entre las comillas se pulsa la tecla de función que queremos utilizar, con lo que aparecerá un símbolo gráfico (el que corresponda a la tecla de función concreta).

Esta es la forma más sencilla e inmediata de programar las teclas de función desde un programa de BASIC. También se puede hacer uso de estas teclas desde programas en lenguaje máquina, pero el sistema de utilización es siempre el mismo: leer el teclado, y si la tecla pulsada es una de las de función, entonces saltar a la parte de código correspondiente, para hacer lo que se haya pensado hacer.

P.: Me gustaría saber que fórmula puede emplearse para que, a partir del número de bloques que ocupa un programa en el *diskette* (número de bloques que se lee en el directorio), pueda saber cuantos *bytes* de memoria ocupa el programa en la memoria del ordenador.

M. Villasante
Madrid

R.: Para hacer una estimación de la cantidad de memoria que utiliza un programa, a partir del número de bloques que ocupa en el *diskette,* hay que dividir este número de bloques por 4, con lo que se obtiene el número de *Kbytes* aproximados que ocupa el programa. También se puede multiplicar el número de bloques por 254, con lo que se obtiene la longitud del programa en *bytes* . Estas fórmulas se derivan del hecho de que cada bloque o sector del *diskette* está constituido por 256 *bytes,* dos de los cuales son utilizados por el ordenador para informaciones sobre el *diskette* y los 254 restantes son los que se emplean para guardar el programa. Así pues, cada bloque o sector son 254 *bytes* de programa o, lo que viene a ser lo mismo, 1/4 *Kbyte.* Por ejemplo, si un programa consume 50 sectores del *diskette,* su longitud es de 50/4 = 12,5K o bien 50 × 254 = 12.700 bytes.

P.: ¿Es posible utilizar los *joysticks* de los ordenadores Atari en el **Commodore-64,** ya que, por lo que he observado, llevan el mismo tipo de conector?

P. Marín
Madrid

R.: Pues sí, quizá sea uno de los pocos aspectos en los que el **Commodore-64** es compatible con otros ordenadores, pues tanto Atari como **Commodore** ha elegido la misma configuración de conexiones para los *joysticks.* Cualquier *joystick* de Atari funciona perfectamente con los ordenadores **Commodore,** tanto con el **C-64** como con el **VIC-20,** sin que haya ningún tipo de problemas.

Y no sólo con Atari, sino, lo que es más importante, con cualquier ordenador MSX, sea de la marca que sea, ya que estos ordenadores también han escogido el mismo tipo de conector y de conexión que utiliza **Commodore** para los *joysticks.* En este aspecto hay que estar agradecidos, ya que hay gran variedad donde escoger. Es una lástima que no pase lo mismo con otros elementos como el *cassette,* la unidad de discos o las impresoras.

P.: Hace poco que tengo un **C-64** y compro su revista; en ella he encontrado programas interesantes y algún que otro quebradero de cabeza. Dado que soy nuevo en el mundo de los ordenadores, mi mayor interés reside en copiar programas para así familiarizarme un poco con el ordenador, y no comprendo como programas «comprobados» pueden fallar en la práctica (no todos, por supuesto). Por ejemplo, el programa «PILOTO DE LAS GALAXIAS» que aparece en su número de octubre, una vez puesto en marcha me indica «ILEGAL QUANTITY ERROR IN 45». La única forma de que esto no suceda es borrando esa línea, pero luego me ocurre lo mismo con la 65. He repasado el programa varias veces y está perfectamente tecleado. Otro tanto ocurre con el programa «ARAÑAS» del número de noviembre, en el que al parecer hay algún problema con las variables en cadena que hay en alguna línea. En fin, que estoy algo decepcionado y he desistido de copiar programas porque después de horas de trabajo el resultado es nulo. He de decirles, sin embargo, que los de «BARCOS» y «PICAS» me funcionaron a la primera. Por favor, díganme si están mal escritos los programas o qué sucede.

J. Molló (Lérida)

R.: Todos los programas que publicamos en **Commodore Magazine** están efectivamente comprobados. El proceso que seguimos, a la hora de publicar los listados, consiste en comprobar que el programa funcione correctamente y sacar el listado por impresora, directamente de la memoria del ordenador, sin ningún tipo de modificaciones que pudieran afectar a su funcionamiento. Posteriormente, dicho listado es fotografiado e impreso en la revista. Con ello conseguimos minimizar el número de errores, pero claro, siempre pueden enredarse las cosas, y nosotros no somos infalibles. El caso es que, repasando los dos programas que citas en tu carta, hemos comprobado que no hay ningún error en ellos. Lamentamos decirte que no vas a tener más remedio que repasarlos otra vez, pero te vamos a dar alguna pista: la línea 45 de «EL PILOTO DE LAS GALAXIAS» incluye la instrucción RED A. Esta instrucción lo que hace es leer un valor de los que aparecen al final del programa, en las sentencias DATA. A continuación aparece un POKE A que se encarga de meter en memoria el valor que se ha leído anteriormente. Esto quiere decir, que aunque la línea 45 esté perfectamente escrita, si hay algún error en las sentencias DATA del final del programa, este error va a aparecer como un error de cantidad ilegal en la línea 45, que es lo que a tí te aparece como «ILEGAL QUANTITY ERROR IN 45». Lo que tienes que hacer es repasar las sentencias DATA, seguro que se te ha escapado alguna coma o tienes algún número muy grande.

En cuanto al programa «ARAÑA», hemos comprobado que tampoco tiene errores y aquí funciona perfectamente. No nos das bastantes datos del error que te aparece pero, por lo que dices, conviene que repases todas las frases que utiliza el programa que vienen almacenadas como «cadenas de caracteres», quizás por ahí se te haya escapado algún error.

Esperamos que con estos consejos, y sabiendo que los dos programas funcionan correctamente y que el listado de la revista es correcto, no tardes en encontrar y corregir las pegas. Si tienes más problemas no dudes en consultarnos.

Commodore 16



por dentro y por fuera

Un prestigioso analista de la industria norteamericana se atrevió, hace un par de años, a vaticinar que cualquier fabricante serio de microordenadores recurriría indefectiblemente a los microprocesadores de 16 bits, si realmente aspiraba a continuar en el mercado. La realidad ha sido bien distinta y la predicción demostró no estar fundamentada. Es más, los microprocesadores de 8 bits han sido revitalizados por los fabricantes de dispositivos semiconductores, que han puesto en el mercado versiones más potentes y rápidas de sus predecesores. Los fabricantes de microordenadores los han acogido con gran entusiasmo, aunque conviene añadir que también utilizan los de 16 bits en microordenadores de la gama alta.

El Vic-20, que en su día fuera la estrella de los microordenadores de bajo precio, le pasa el testigo, como si de una carrera tecnológica de relevos se tratase, al recién anunciado **Commodore 16**. Se trata de un digno sucesor, como veremos a lo largo de la presente descripción.

El **C-16**, le llamaremos así a partir de ahora, es una máquina basada en torno al microprocesador 7501, que es descendiente directo del popular **6502**, empleado por el **Vic-20**. No se trata

de tecnología punta, pero resultará muy familiar a quienes hayan trabajado con el código máquina en el **Vic-20** o el **C-64**.

La primera mejora que encontramos son 16 *Kbytes* de memoria RAM (de ahí su nombre) en lugar de los 5K del **Vic**.

La memoria ROM es también de mayor dimensión: 20 K en **Vic** y 32 Kbytes en el **C-16.**

Para los amantes de las cifras proporcionamos ésta a título de curiosidad: La frecuencia de reloj empleada por el microprocesador son 1,76 MHz.

Pero veamos lo que pueden contener los 32 K de ROM de este nuevo vástago de **Commodore**. Como es habitual, comprende el intérprete del lenguaje BASIC, así como las rutinas gráficas y el sistema operativo adecuado para el correcto funcionamiento del ordenador. Las capacidades gráficas de esta nueva máquina nos han sorprendido gratamente.

La carcasa utilizada es la misma empleada por los modelos **Vic-20** y **C-64**, por lo cual su aspecto es prácticamente igual a simple vista, pero una mirada más minuciosa revela novedades en el teclado, que resalta sobre el fondo oscuro de la carcasa. Las teclas siguen manteniendo la misma calidad que en sus predecesores. De hecho, opinamos que es uno de los teclados de mejor calidad en ordenadores de esta gama de precios.

Por lo pronto, las teclas de movimiento del cursor han sido multiplicadas por dos en número. Cada uno de los cuatro posibles movimientos del cursor por la pantalla dispone de tecla propia, que pasa a sustituir a la combinación de solamente dos teclas más SHIFT, tradicional en **Commodore**. No obstante tenemos algo que objetar, la disposición de las cuatro teclas no parece muy afortunada en lo que a comodidad de utilización se refiere. Es indudable que su ubicación en una disposición más cómoda habría alterado el diseño externo de la carcasa. Tal vez el incremento en los costos de producción no le pareció oportuno al fabricante. De todas formas la mejora es de tener en cuenta.

Siguiendo la tradición establecida, una gran parte de las teclas son polivalentes; es decir, sirven para una cosa u otra, según se las utilice en solitario o en combinación con otras teclas especiales, como son SHIFT y la que lleva impreso el logotipo de **Commodore**.

La tecla CTRL (control) persiste igualmente. La combinación de ella o la tecla **Commodore** con las destinadas a los números 1 a 8 nos permiten la elección de los diversos colores disponibles en la «paleta» del **C-16**. La combinación de CTRL con otras dos teclas nos permiten decidir que los caracteres parpadeen o dejen de hacerlo en la pantalla. La combinación con la S posibilita hacer una pausa durante el listado de líneas en pantalla.

ESC es una nueva tecla que aparece en el **C-16** y que proporciona funciones adicionales al ordenador, cuando se la utiliza en compañía de una tecla alfabética, tal como se ve en el cuadro 1.

Las cuatro teclas de función, situadas en hilera a la derecha del teclado, también se conservan. Son bivalentes en combinación con la tecla SHIFT. En el **Vic-20** (y el **C-64**) no tenían una utilidad definida a priori. Sin embargo, en el **C-16** proporcionan nuevas posibilidades en cuanto conectamos la alimentación al ordenador. F1 dispone la entrada en operación de uno de los modos gráficos. F2 imprime en la pantalla DLOAD", con lo cual solamente nos queda teclear el nombre del programa que deseamos cargar en la memoria del ordenador a partir del *diskette*. F3 nos proporciona un listado del directorio de los ficheros almacenados en un *diskette*, evitando así la tediosa tarea de utilizar los comandos OPEN y LIST, necesarios en el **Vic-20** para conseguir esto mismo. F4 limpia la pantalla inmediatamente, sea cual sea el modo gráfico en el que estamos trabajando. Es una característica que sabrán apreciar los usuarios del **C-64** que hayan trabajado con los gráficos de alta resolución. F4 a F6 son accediadas mediante la combinación de F1 a F3 con SHIFT.

F5 imprime en pantalla DSAVE", de tal manera que solamente precisamos escribir el nombre bajo el cual deseamos cargar el fichero en un *diskette*. F6 hace lo mismo que si hubiéramos escrito RUN y presionamos Return.

La última tecla de función lleva escrito HELP (ayuda) en la parte superior y F7 en la frontal. HELP se convierte en una ayuda extremadamente



práctica, lista en pantalla la línea en la que se ha producido un error cuando utilizamos el BASIC, indicando dónde se ha producido el mismo . En el modo texto, el **C-16** produce un formato de pantalla de 25 líneas de hasta 40 caracteres cada una y se puede elegir entre una gama de 16 colores, para diseñar la visualización con los que más nos gusten.

En el modo de alta resolución la pantalla se transforma en una matriz compuesta por 200 puntos en sentido vertical y 320 en el horizontal, disponiendo de dos colores para cada cuadrado de 8 por 8 puntos. Al pasar a otro cuadrado de 8 por 8 eso dos colores pueden ser distintos. También es posible disponer de cuatro colores, en el llamado modo multicolor, pero la resolución se divide por dos en el sentido horizontal, esto es, 200 por 160 puntos.

Démosle, ahora, la vuelta al ordenador para observar su parte posterior. Vista de izquierda a derecha, lo primero que observamos es una amplia ranura sobre la que se puede leer *Memory expansion*. Se trata de un *port* destinado a alojar los distintos cartuchos de *software* en ROM que vayan apareciendo en el mercado. Sin embargo, los cartuchos existentes para el Vic no pueden conectarse aquí.

A continuación aparece el clásico conector destinados al cable coaxial de conexión a la antena de un televisor en color doméstico. A través de esta salida se obtiene la señal de vídeo que modulan a una señal de UHF, de tal manera que puede atacar directamente a la entrada de antena del receptor. No obstante, también es posible aplicar la señal directamente a un monitor, blanco y negro o color, para obtener una imagen de mayor calidad. El conector empleado en este caso es del mismo tipo que los empleados en **Vic-20** y **C-64** para idéntico fin.

La conexión con los dispositivos periféricos se efectúa mediante el mismo conector empleado anteriormente en los otros dos modelos referidos. Se trata de un *interface* serie, similar al RS-232, pero que responde a un estándar de **Commodore**, de tal manera que con el **C-16** pueden trabajar la unidad de *diskettes* **1541**, la impresora *plotter* **1520** y la impresora matricial **1526**, así como los nuevos **SFS481**, que es la nueva unidad rápida de *diskettes* (a la que estamos deseando hincar el diente), la impresora matricial de color **MCS 601**, la matricial **MPS 802** o la de margarita y mayor calidad de impresión **DPS 1101**.

El surtido de conectores posterior termina con el destinado a la unidad de *cassettes* **Datassette 1531**, similar al **1530** pero, según el fabricante, con características mejoradas. Como sucede con otros periféricos y accesorios, el **Datassette** antiguo no es directamente conectable al **C-16**, el conector es distinto. Mintras el **1530** lo lleva de forma alargada, en el **1531** se ha optado por uno redondo.

Las sorpresas no acaban. El lado derecho de la carcasa muestra una ventana alargada que acoge, en primer lugar, a los dos conectores reservados a otros tantos *joysticks*. Su configuración nada tiene que ver con los tipo D de 9 contactos utilizados en **Vic** y **64**. Por tanto los *joysticks* de aquéllos no son directamente conectables al nuevo modelo.

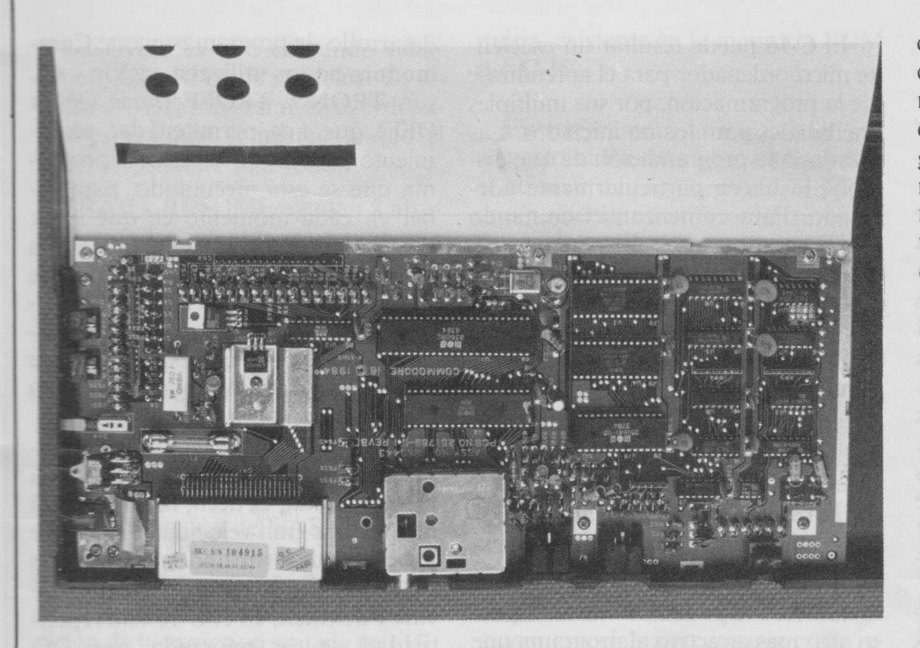
A continuación una novedad, algo que se echaba en falta: un botón de puesta a cero del sistema por lo rápido (RESET). Se trata de un pulsador más o menos camuflado para evitar disgustos involuntarios. Cuando se le presiona la pantalla es borrada y se evapora el programa en BASIC que tuviéramos en la memoria RAM. Por otro lado, la combinación de este botón con la tecla Run/Stop posibilita que saltemos inmediatamente al monitor de código máquina.

En este flanco derecho solamente nos queda por describir el interruptor de paso de alimentación eléctrica y el conector destinado al cable que procede del módulo externo de alimentación, que toma energía de la red eléctrica. Esta vez el conector empleado también es diferente al homólogo de **Vic-20** y **C-64**.

Aparte de las características de nuevo cuño descritas, lo que realmente hace del **C-16** una máquina excepcional es su versión del BASIC, la 3.5, a nuestro juicio la mejor y más versátil de que dispone **Commodore**. En muchos aspectos es similar al BASIC del **8032**, pero además se añaden los

CUADRO 1

ESC +
- A: Inserción automática.
- B: Ajusta la esquina inferior derecha de una ventana en pantalla.
- C: Cancela la inserción automática.
- D: Borra la línea en que estamos.
- I: Inserta una línea.
- J: Nos sitúa en el comienzo de una línea en la que estamos trabajando.
- K: Nos lleva al final de la línea en la que estamos situados.
- L: Activa la pantalla.
- M: Desactiva la pantalla.
- N: Nos devuelve al formato de pantalla normal.
- O: Cancela los modos de inserción, comillas, inverso y parpadeante.
- P: Borra todo lo que haya desde el principio de la línea hasta el cursor.
- Q: Borra todo lo que haya desde el cursor hasta el final de la línea.
- R: Reduce la visualización en pantalla.
- T: Ajusta la esquina superior derecha de una ventana en la pantalla.
- V: Produce un *scrolling* hacia arriba.
- W: Produce el *scrolling* hacia abajo.
- X: Anula cualquier función previamente accedida con ESC.



comandos tan echados de menos para una más fácil utilización de las posibilidades gráficas y sonoras del ordenador.

Una vez que hemos conectado todos los elementos y cables necesarios para que el **C-16** funcione con propiedad, enchufamos la unidad de alimentación a la red eléctrica y ponemos en ON el interruptor de puesta en marcha. En la pantalla del televisor (o monitor) es visualizado el mensaje de saludo que identifica al sistema. En primer lugar nos hace saber que tenemos a nuestra disposición la referida versión 3.5 del BASIC de **Commodore**. A continuación nos indica que disponemos de 12277 bytes libres a nuestra disposición, y por último, el mensaje READY pone el ordenador a nuestras órdenes mediante el cursor parpadeante que aparece en la línea de más abajo.

Otro detalle más del cuidado diseño del **C-16**: podría ocurrir que nuestro televisor estuviera ligeramente desajustado, no permitiéndonos ver los 40 caracteres máximos que en modo texto pódemos visualizar en una línea. Lo más lógico consisitiría en recortar los extremos laterales. Pues bien, ESC más la tecla R reducen la longitud máxima a 38 caracteres por línea.

Algo que los usuarios y conocedores del **C-64** echarán de menos es la ausencia de los *Sprites*, sencillamente porque no han sido previstos en el nuevo modelo. Francamente una pena.

Los comandos gráficos no tienen desperdicio. Como sucede con los modelos anteriores, la pantalla del **C-16** se divide en tres partes: el reborde o marco, el fondo y los caracteres. Cada una de ellas puede disponer de su propio color. Con la ayuda de un nuevo comando, con el significativo nombre de COLOR, podemos definir uno entre los 16 colores posibles para cada parte. Por ejemplo, COLOR 0,3 pone el fondo sobre el que aparecen los caracteres en color rojo y COLOR 1,7 hace que éstos tomen el color azul. Este comando puede utilizarse en modo diferido; es decir, formando parte de una línea de programa.

Pero lo realmente atractivo es el nuevo conjunto de comandos destinados a facilitar la utilización de las capacidades del ordenador para crear bonitos gráficos, de manera más sencilla que antes. El comando GRAPHIC nos permite elegir entre distintas posibilidades, por medio de los dos parámetros numéricos que pueden seguirle. En un principio podemos optar por uno entre cinco modos, que son: sólo texto en pantalla, gráficos en alta resolución y gráficos multicolor por un lado y otros dos más que introducen el concepto de pantalla partida, donde la pantalla se divide para proporcionarnos simultáneamente texto y gráficos de alta resolución o multicolor. En estos dos últimos modos de funcionamiento, la parte inferior de la pantalla aloja hasta a cinco líneas de texto. Sin embargo esto no representa una novedad absoluta en los ordenadores de **Commodore**. Sin ir más lejos, en el **C-64** se puede obtener algo similar, pero es necesario disponer de un profundo conocimiento de lenguaje máquina. Lo novedoso es el comando en sí, accesible desde el BASIC. Este mismo comando sirve para limpiar el contenido de la pantalla al mismo tiempo que definimos el modo de trabajo, pero existe otro nuevo comando, SCNCLR que borra la pantalla sin alterar el modo de trabajo.

Es posible también dibujar rectángulos o círculos en la pantalla de manera sencilla. BOX nos dibuja un rectángulo de dimensión marcada por los parámetros que escribamos a continuación, además de poder decidir un ángulo de rotación con respecto a la posición inicial del rectángulo. Por otro lado, puede aparecer en forma de cuatro líneas que lo conforman o como una superficie coloreada.

CIRCLE sirve para que aparezca un círculo de centro en un lugar que elegimos y un radio también marcado por nosotros. Añadiendo nuevos parámetros, CIRCLE nos permite trazar elipses, arcos, elipses desfasadas entre sí un número dado de grados y hasta polígonos tales como un triángulo, un cuadrado, etc.

El comando PAINT nos ofrece la posibilidad de pintar; esto es, rellenar con cualquiera de los colores un área cerrada, definida por trazos en la pantalla.

Igualmente podemos hacer cosas más sencillas con un solo comando.

DRAW (dibujar) pone a nuestro alcance la posibilidad de dibujar puntos o líneas que unen a dos puntos entre sí. CHAR nos permite escribir cualquier leyenda a partir de un lugar determinado de la pantalla.

El sonido sigue estando presente entre las característicos del **C-16**. A **Commodore** siempre le hemos reprochado que no haya incorporado nunca comandos para utilización directa del sonido desde el BASIC. Las posibilidades de generación de sonidos no son tan flexibles y complejas en este modelo como lo son en el **64**. En realidad son dos comandos y dos generadores de lo que dispone este nuevo ordenador. VOL seguido por un número entre 0 y 8 ajusta el volumen final de la señal sonora.

El comando SOUND es más versátil, va seguido por tres parámetros que definen si queremos trabajar con el primer o segundo generador de sonido, la frecuencia de la nota y la duración de la misma. Decimos que el **C-16** dispone de dos generadores de sonido, el primero sólo produce tonos y el segundo proporciona tonos o ruido blanco, a elegir.

El **C-16** puede resultar un excelente microordenador para el aprendizaje de la programación, por sus múltiples facilidades para los no iniciados. Las ayudas a la programación de que dispone le hacen particularmente adecuado. Para comenzar, el comando AUTO escribe automáticamente el siguiente número de línea cada vez que apretamos la tecla Return, quedando el incremento entre números de línea definido por el parámetro que escribimos detrás del comando.

DELETE borra la línea o conjunto de líneas de programa que especifiquemos tras él.

HELP nos ayuda a localizar dónde se ha producido un error en nuestro programa.

También le podemos dar un aspecto algo más atractivo al programa que acabamos de producir, en el cual los números de línea no vayan a incrementos regulares. RENUMBER renumera las líneas del mismo a partir de una determinada, siguiendo un incremento que también establecemos nosotros. Otro par de comandos que resultan de tremenda ayuda durante el desarrollo de programas, y que **Commodore** no los utilizaba en **Vic** y **64**, son TRON y TROFF (*trace* ON u OFF) que nos permiten dar seguimiento (o dejar de hacerlo) al programa que se está ejecutando, para saber en cada momento en qué lugar nos encontramos, pues en la pantalla aparece impreso el número de línea de cada una de las que van ejecutándose, en orden. De esta manera, si el programa no hace lo que esperamos que hiciera es más fácil saber dónde falla el proceso.

El BASIC es un lenguaje de programación cuya principal característica es la iteratividad, es decir, la posibilidad de repetir a gran velocidad un tipo de proceso determinado. El bucle FOR...NEXT es un claro ejemplo de ello. Pues bien, en el **C-16** esta repetitividad se ve reforzada por DO (LOOP) WHILE (UNTIL EXIT), que ejecuta las sentencias situadas entre las que contienen a DO y LOOP. Junto con ellos se puede utilizar UNTIL o WHILE para continuar o salir del bucle en función de que se cumpla una cierta condición establecida por el programador.



El RESET es una de las novedades que incorpora el C-16.

GETKEY es otro comando echado en falta en otros modelos. Es similar a GET, pero esta vez el ordenador espera por sí mismo a que el usuario presione una tecla.

TRAP y RESUME ayudan a atrapar errores e intentar reejecutar una línea en la que se supone hay error. Ambos comandos pueden resultar muy útiles durante el desarrollo de un programa.

Saber cómo un número escrito en hexadecimal se escribe en decimal o viceversa es tarea de las nuevas funciones HEX$ y DEC.

En esta versión del BASIC el clásico IF...THEN se ve enriquecido por ELSE, que hace lo que establezcamos a continuación si no se cumple la condición de la expresión situada tras IF.

JOY es una función numérica relacionada con los *joysticks*. Así, para leerlos ya no es necesario recurrir a los PEEK habituales en los otros modelos.

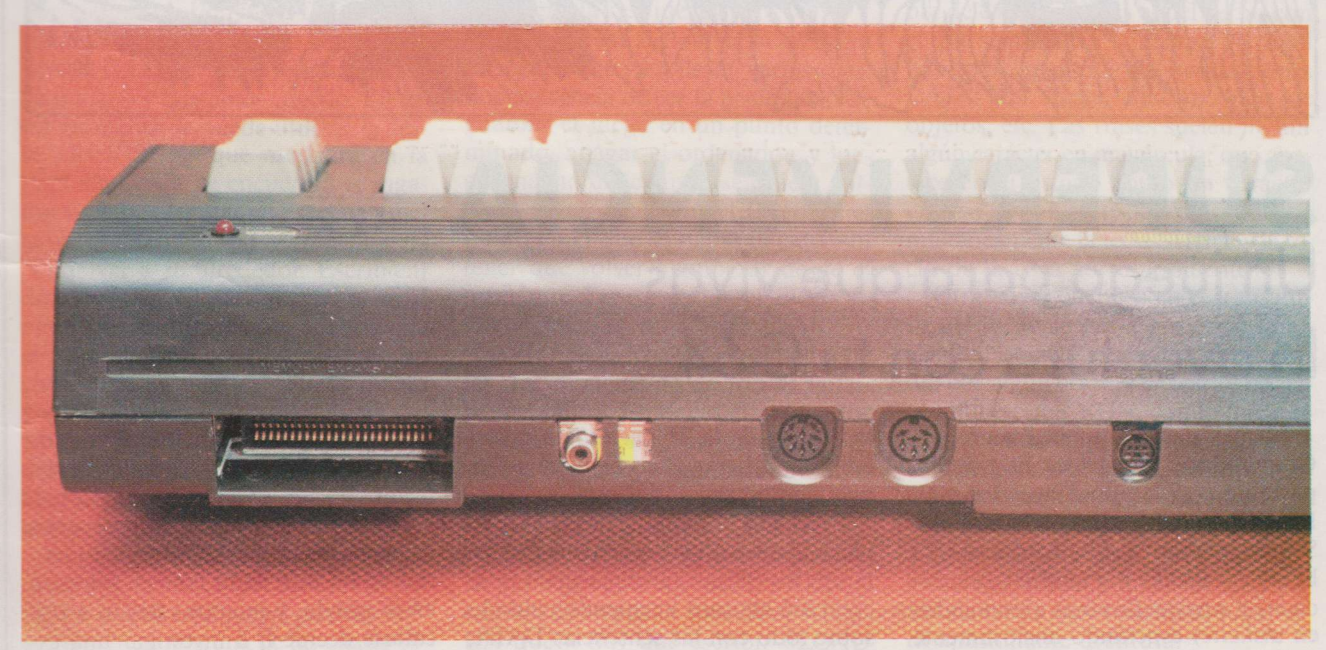
El comando MONITOR nos saca del BASIC y pone a nuestro alcance un programa monitor de código máquina, existente en la memoria ROM del **C-16.**

Esto representa una novedad con respecto al **Vic-20** y al **C-64,** que precisaban de un programa externo para que pudiéramos utilizar un monitor de estas características. Con su ayuda podemos desarrollar más fácilmente nuestros programas escritos en código máquina. Podemos volcar directamente en pantalla el contenido de una determinada sección de la memoria, ensamblar y desensamblar código, transferir o comparar partes de la memoria, entre otras varias posibilidades.

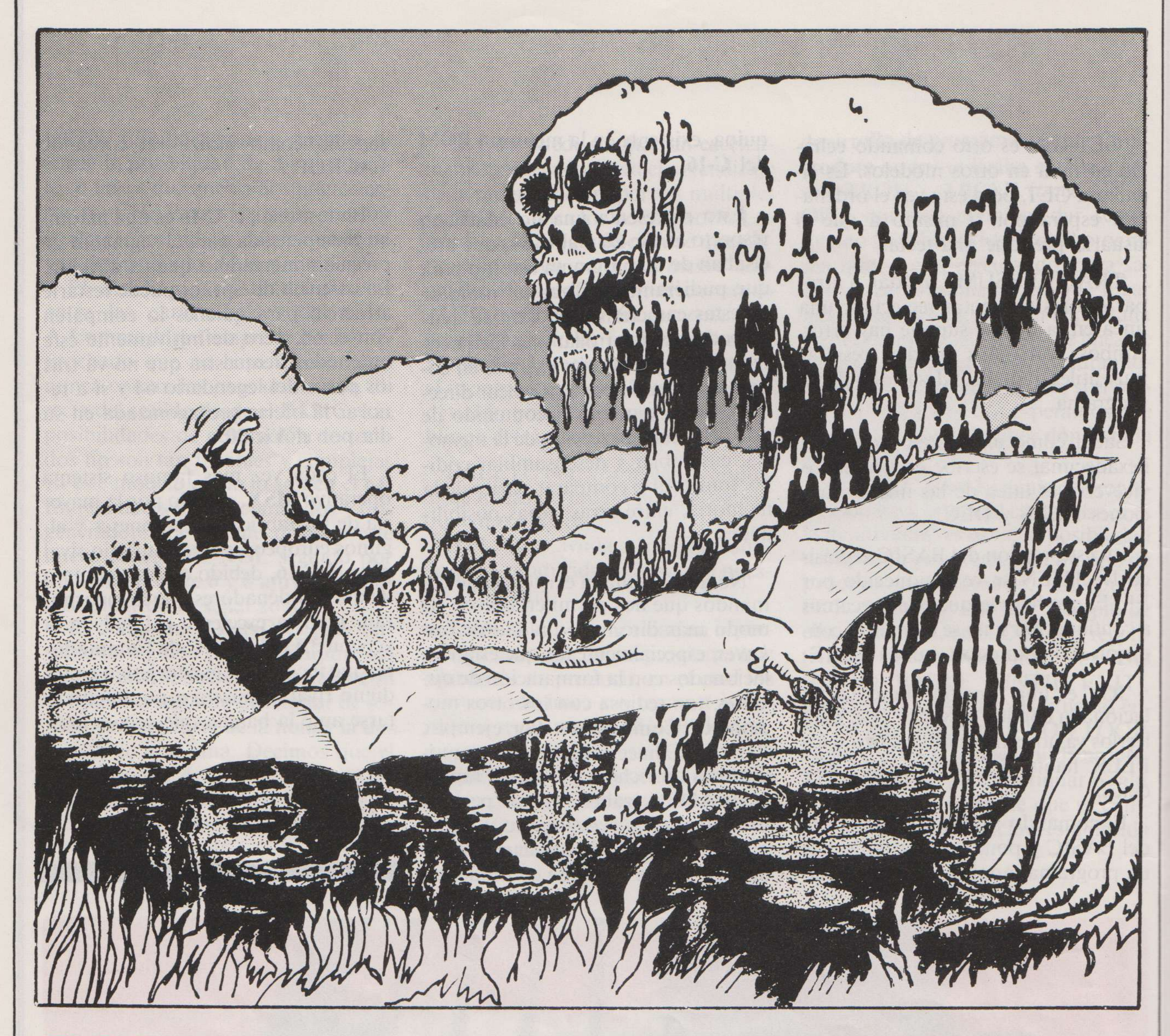
Notable resulta la existencia de comandos que nos permiten utilizar de modo más directo la unidad de *diskettes,* especialmente los que están relacionados con la formatación de *diskettes,* tan tediosa con los otros modelos de **Commodore.** Por ejemplo, RENAME hace posible que rebauticemos a un fichero por la vía rápida y HEADER (cabecera) nos permite formatear un *diskette* nuevo. Las copias de seguridad de un determinado *diskette* se consiguen de un modo tan sencillo como utilizar el comando BACKUP.

En resumen, el **C-16** es una máquina bien pensada para el segmento de precios y mercado a que va dirigida. La carencia de Sprites puede restarle atractivo para quienes lo comparen con el **64,** pero definitivamente este ordenador demuestra que no va tras los pasos del legendario **64** y sí a renovar la oferta protagonizada en su día por el **Vic-20.**

El cada vez más famoso sistema operativo **MSX,** elegido por la mayoría de los fabricantes japoneses y algunos europeos, parece un serio rival para el **C-16,** debido al gran número de microordenadores que lo incorporan, pero la excelente calidad de la versión 3.5 del BASIC de **Commodore,** que incorpora, le convierten en un digno rival dispuesto a no amedrentarse ante la batalla que se avecina.



Conexiones posteriores del Commodore 16.



SUPERVIVENCIA (y 2ª parte)

Un juego para que vivas la aventura con tu C-64

El mes pasado publicábamos, en Commodore Magazine, la primera parte de esta fabulosa aventura SUPERVIVENCIA, destinada al C-64. Seguro que muchos de vosotros ya la habréis tecleado y la tendréis guardada en el *cassette* (los más afortunados en *diskette*) esperando ansiosamente la llegada de la 2ª parte para poder empezar a jugar. Pues bién, ya no tenéis que esperar, aquí está.

El listado es bastante largo, así que paciencia y copiarlo poco a poco evitando los errores (aunque siempre se escapará alguno) y teniendo cuidado sobre todo con las sentencias DATA, que suelen ser la principal fuente de problemas, y muchos, difíciles de detectar.

En la primera parte explicábamos un poco la trama del juego. El jugador se ha estrellado con su avioneta en la jungla, y mientras llegan a rescatarle decide ir a investigar una ex-

Rutinas de supervivencia

406-420 *Rutina de TROCEAR. Es algo diferente a la rutina de cortar. La primera supone que eres un poco manazas, y que trocear no es lo adecuado. Por el contrario, cortar ya supone una mayor habilidad, por lo que puede dar mejores resultados.*

422-476 *Rutina de ENCENDER. Este verbo significa «prender fuego a» y a lo largo del juego habrá que utilizarla en algunas ocasiones para poder seguir adelante. El comando (encender) sólo se ejecutará cuando se cumplan determinadas condiciones, por ejemplo habrá que pensar en llevar cerillas.*

478-526 *Rutina de TREPAR. Se utiliza para subir a algún sitio, ya sea una escalera, o un árbol o cualquier otro elemento del juego.*

528-538 *Rutina de ATACAR. A veces, cuando las cosas se pongan feas, puede intentarse atacar a algo o a alguien, pero como norma general, podemos decir que la violencia no conduce a ningún sitio.*

540-548 *Rutina de MATAR. Desde luego que esto de matar es una cosa muy seria y no es para andar jugando con ella.*

550-564 *Rutina de GOLPEAR. No sólo se trata de golpear a alguien, lo que se puede hacer en cualquier momento, sino de golpear algún objeto, por ejemplo un candado, con la intención de romperlo y poder seguir adelante.*

566-626 *Rutina de HACER. Puedes hacerte o construirte objetos, a veces no vas a tener más remedio. Los materiales con que se hace el objeto son decrementados o escondidos, y se ajustan ciertos indicadores o flags para asegurarse de que sólo se hacen un cierto número de objetos a partir de un ingrediente básico. En general, para hacer los objetos habrá que tener primero todos los materiales necesarios.*

628-634 *Rutina de ACEITAR. Es una forma un tanto rara de decir lubricar, pero significa lo mismo, esto es: dar o echar aceite a algo o a alguien. Puede que alguna puerta que se resista a abrirse esté necesitando un poco de aceite.*

636-644 *Rutina de APUÑALAR. La falta de «Ñ» en el ordenador hace que haya que escribir «apunalar», pero no importa, la máquina lo entiende. Sobre esta acción decimos lo mismo que sobre la de atacar: que la violencia nunca es buena.*

traña luz que vio desde los aires. Para ello el jugador tiene que ir escribiendo frases en castellano diciéndole al ordenador qué es lo que quiere hacer. Estas frases tienen que ser de la forma «verbo en infinitivo nombre» por ejemplo; «IR NORTE» para dirigirse hacia el norte o «COGER MARTILLO» si vemos algún martillo a mano y queremos cogerlo. En todo momento se puede utilizar la frase «REVISAR» que mostrará en la pantalla donde se encuentra el jugador (que puede estar en una cualquiera de 84 posiciones) y qué objetos puede ver a su alrededor. Escribiendo «INVENTARIO» aparecerá en la pantalla una lista de los objetos que el jugador lleva consigo. Este podrá coger y dejar cuantos objetos quiera y en cualquiera de las posiciones del juego. En la tabla 1 damos una lista de todos los objetos que pueden ir apareciendo a lo largo del recorrido por la jungla.

Otra cuestión interesante que queremos comentar es que en determinadas posiciones del juego ¡no hay salida directa en ninguna de las cuatro direcciones N, S, E y O! por lo que el jugador tendrá que emplear su ingenio y hacer alguna cosa que le permita seguir adelante. Por ejemplo, abrir una puerta o quemar algo o romper algún objeto. Aquí es donde está toda la gracia del juego.

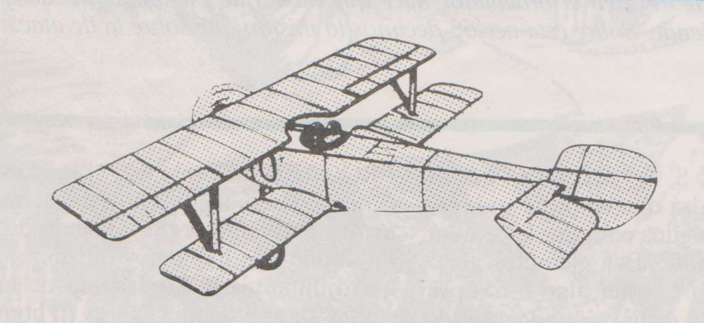
Como llegar al final de la aventura es seguro que va a llevar varios dıas, Supervivencia ofrece la posibilidad de guardar el juego en un punto determinado, apagar el ordenador, y luego, más tarde, cuando se quiera volver a jugar, recuperar el juego en el punto en el que se había dejado, sin tener que volver a empezar desde el principio. Para guardar el juego se puede escribir «ABANDONAR» o «SAVE PROGRESO», cualquiera de las dos maneras de decirlo es igualmente válida. Entonces el ordenador preguntará si estamos utilizando *cassette* o *diskette* y, después de responderle, guardará el juego en el punto en el que se encuentra. Para recuperar la posición en el juego, después de cargar el programa y escribir RUN, y cuando aparezca la frasecita «AHORA qué?» hay que escribir «LOAD PROGRESO» con lo que el juego se cargará desde la cinta o desde el *diskette* y podremos seguir jugando desde donde nos habíamos quedado.

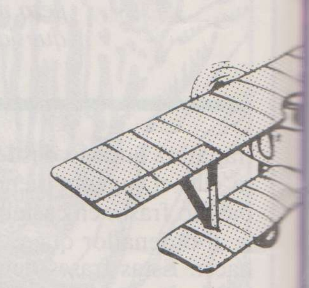
Adjuntamos la explicación de las rutinas de esta segunda parte. El nombre de cada rutina es el del verbo con el que se accede a la misma. Al final hay toda una serie de instrucciones DATA con frases, descripciones de objetos, etc. Las frases suelen incluir algún carácter en mayúscula, que aparece en el listado como un carácter gráfico y que se teclea pulsando simultáneamente la tecla SHIFT y la tecla de la letra correspondiente.

Otra advertencia, si en el transcurso del juego se pulsa por error la tecla RUN/STOP o se para el programa por algún error al teclearlo, no hace falta escribir RUN, con lo que tendríamos que empezar el juego desde el principio, sino que se puede arrancar en el punto en que estábamos sin más que escribir GO TO 98.

Sólo nos queda recomendaros paciencia al teclear y desearos un montón de horas entretenidas al jugar esta apasionante aventura.

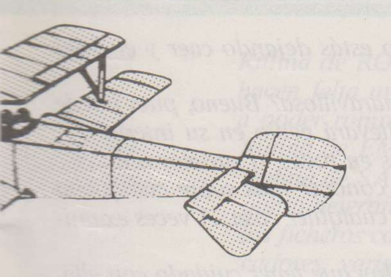
```
406 IFNO=0THEN690
408 GOSUB856
410 IFOB%(19)=-1THENPRINT"/O SE ØTROCEAR■ CON UN MACHETE!":GOTO100
412 IFOB%(24)<>-1THENPRINT"-ON QUE?":GOTO100
414 IFNO<>40THENPRINT"/O TIENE NINGUN SENTIDO!!":GOTO100
416 IFBT=1ANDOA=1THENPRINT"●E ACABO LA MADERA!":OB%(40)=0:GOTO100
418 PRINT"|ROCEAR MADEROS PUEDE QUE TE DIVIERTA,AMI NO!. ¬IENSA EN HACER ALGO ";
420 PRINT"CON ELLOS, POR FAVOR!":GOTO100
422 IFNO=0THEN690
424 GOSUB856
426 IFNO<>22THEN434
428 IFOB%(22)<>-1THENPRINT"/O LO LLEVAS, LISTO!":GOTO100
430 IFOB%(22)=-1THENOB%(23)=-1:PD=0:PRINT" | SE HIZO LA LUZ!"
431 IF OB%(22)=-1THENOB%(22)=0:FORTP=0TO1000:NEXTTP:GOTO98
432 IFOB%(23)<>0THENPRINT" |A ESTA ENCENDIDA, TONTO!":GOTO100
434 IFOB%(13)<>-1THENPRINT"-OMO? _ROTANDO DOS PIEDRAS?":GOTO100
436 IFNO=5THEN444
438 IFNO=12THEN458
440 IFNO=42THEN464
442 PRINT"LO QUE DICES NO TIENE SENTIDO!":GOTO100
444 IFCP<>21THENPRINT"¯STAS DEMASIADO LEJOS!":GOTO100
446 PRINT"¬L EDIFICIO ARDE COMO SI FUERA DE MADERALAS CENIZAS SE ESPARCEN ";
448 PRINT"CON EL VIENTO.  ¬L CALOR RAJA EL CAMINO DE BARRO QUE SE CONVIERTE ";
450 PRINT"EN POLVO.ØØØØØ":CP=22:P%(21,2)=0:OB%(5)=0
```

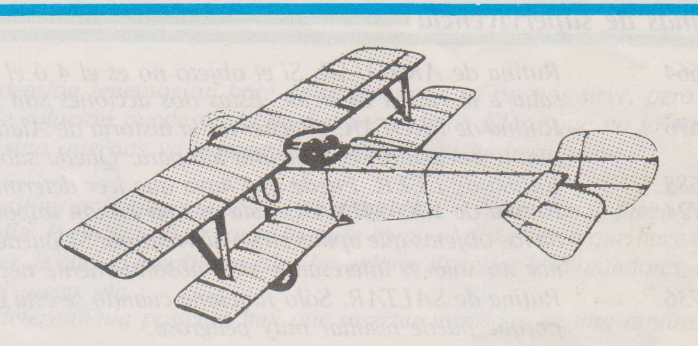




```
452 FORR=1TOLO:IFOB%(R)=21THENOB%(R)=22
454 NEXT
456 GOSUB56:GOTO100
458 IFCP<>38THENPRINT"¬L ACEITE ARDE BIEN, PERO NO PASA NADA.":GOTO100
460 PRINT"¬L CALOR DE LA LLAMA FUNDE AL FANTASMA, DEJANDO EL PASO LIBRE."
462 P%(38,3)=39:OB%(35)=0:OB%(6)=0:GOTO100
464 IFCP<>20ORCL<>1THENPRINT"/O SIRVE HASTA QUE NO SEPAS DONDE VAS.":GOTO100
466 IFHE<>0THENPRINT" |A LO HAS HECHO ANTES!":GOTO100
468 PRINT"¬L SETO SE QUEMA SOLTANDO CHISPAS.":IFWA<>1THENPRINTEX$:GOSUB1290
470 PRINT"¬L FUEGO SE APAGA FINALMENTE MOSTRANDO  UN CAMINO AL SUR."
472 IFWA<>1THENPRINTPL$:FORI=1TOLO:IFOB%(I)=2THENOB%(I)=0
474 IFWA<>1THENNEXT
476 P%(2,1)=5:HE=1:OB%(1)=2:GOTO100
478 IFNO=0THEN690
480 GOSUB856
482 IFCP=2THEN492
484 IFCP=13THEN500
486 IFCP=22ORCP=23THEN512
488 IFCP=57ORCP=60THEN520
490 PRINT"/O HAY NADA INTERESANTE QUE TREPAR!":GOTO100
492 IFNO<>43ANDNO<>44THENPRINT"/O SEAS TONTO!":GOTO100
494 PRINT"¯ESDE LO ALTO DEL ALA PUEDES VER POR EN-CIMA DEL SETO.♠L SUR HAY UN ";
496 PRINT"CAMINO HACIAEL BOSQUE;EN OTRAS DIRECCIONES LA JUNGLA ES IMPENETRABLE."
498 PRINT"|AJAS DEL ALA.":CL=1:GOTO100
```

```
500 IFNO<>45ANDNO<>46THENPRINT"¬ARA QUE?":GOTO100
502 IFOB%(18)<>-1THENGOSUB1290:GOTO100
504 PRINT"¬ESDE LO ALTO DEL ARBOL DIVISAS EL PANO-RAMA.♠ LO LEJOS VES UN ";
506 PRINT"PENACHO DE HUMO  QUE SE ELEVA DESDE UN VOLCAN.♠L OESTE SEDIVISA UNA ";
508 PRINT"COLINA CONICA, EN CUYA CIMA  HAY UN EDIFICIO.";
510 PRINT"|AJAS DEL ARBOL.":CL=2:GOTO100
512 IFNO<>41THENPRINT"|OTALMENTE RIDICULO!":GOTO100
516 IFCP=22THENCP=23:GOTO98
518 IFCP=23THENCP=22:GOTO98
520 IFNO<>3THENPRINT"¬H?":GOTO100
522 IFOB%(3)<>CPTHENPRINT"/O ESTA AQUI!":GOTO100
524 IFCP=57THENCP=60:OB%(3)=CP:GOTO98
526 IFCP=60THENCP=57:OB%(3)=CP:GOTO98
528 IFNO=0THEN690
530 GOSUB856
532 IFNO=37THENGOSUB1286:GOTO100
534 IFNO=39THENPRINT"|E QUEMAS LA MANO, PERO NO LE MUEVES.":GOTO100
536 IFNO=35THENPRINT"|U MANO LE ATRAVIESA.":GOTO100
538 PRINT"\ENUDA TONTERIA ES ESO!":GOTO100
540 IFNO=0THENPRINT"\ATAR A QUIEN?":GOTO100
542 GOSUB856
544 IFNO=37THENGOSUB1286:GOTO100
546 IFNO=39THENPRINTBU$:GOTO100
548 PRINT"♠ LOS DIOSES NO LES IBA A GUSTAR, ASI  QUE OLVIDALO.":GOTO100
```





```
550 IFNO=0THENPRINT"|OLPEAR QUE?":GOTO100
552 GOSUB856
554 IFNO=37THENGOSUB1286:GOTO100
556 IFOB%(9)<>-1THENPRINT"|E HACES PUPA, PERO NO OCURRE NADA.":GOTO100
558 IFCP<>50THENPRINT"¬SO ES UNA TONTERIA!!":GOTO100
560 IFNO=57THENPRINTBO$:BO=1:OB$(7)="UNA VERJA CON CANDADO"
562 IFPA=1THENPRINTGA$:GR=1:P%(50,2)=51:OB$(7)="UNA VERJA ABIERTA":GOTO100
564 IFPA=0THENPRINT"¬L CANDADO LA MANTIENE CERRADA.":GOTO100
566 IFNO=0THENPRINT" |ACER QUE?":GOTO100
568 GOSUB856
570 IFNO=32THEN602
572 IFNO=16THEN590
574 IFNO=30THEN606
576 IFNO=3THEN582
578 IFNO=31THEN618
580 PRINT" |A ESTA HECHO!":GOTO100
582 IFOB%(9)<>-1OROB%(24)<>-1THENPRINT"/O TIENES HERRAMIENTAS.":GOTO100
584 IFOB%(29)<>-1ANDOB%(8)<>-1THENPRINT"|E FALTAN MATERIALES.":GOTO100
586 PRINT"Γ.⁄.":OB%(3)=CP:OB%(29)=0:OB%(8)=0:PRINT" |A TIENES ESCALERA":ZZ=ZZ-1
588 GOTO100
590 IFOB%(32)<>-1ANDOB%(14)<>-1THENPRINT"_ALTA CUERDA!":GOTO100
592 IFOB%(15)<>-1THENPRINT"/O HAY IMAN! |USCA!!":GOTO100
594 PRINT"Γ.⁄. |IENES UN COMPAS HECHO A MANO.":OB%(16)=-1:OB$(14)=""
596 ZZ=ZZ-1:OB$(17)="UN GIROCOMPAS":OB%(15)=0:OB%(14)=0
```

```
598 IFOB%(32)<>0THENOB%(32)=0
600 GOTO100
602 IFOB%(16)<>-1THENPRINT"|E HACE FALTA CUERDA.":GOTO100
604 OB%(16)=0:OB%(32)=-1:OB%(15)=CP:PRINT"|A TIENES UN LAZO":GOTO100
606 IFOB%(8)<>-1THENPRINT"¯OMO VAS A SUJETAR LOS MADEROS?":GOTO100
608 IFOB%(40)<>-1THENPRINT"¯ONDE ESTAN LOS MADEROS?":GOTO100
610 IFOB%(9)<>-1THENPRINT"XAS A CLAVAR CON LA MANO?":GOTO100
612 PRINT"/N BUEN BOTE DE MADERA.":BT=1:OB%(30)=CP:OB%(8)=0
614 IFBT=1ANDOA=1THENPRINT"♥E ACABO LA MADERA!":OB%(40)=0
616 GOTO100
618 IFOB%(40)<>-1THENPRINT"_ALTA MADERA.":GOTO100
620 IFOB%(24)<>-1THENPRINT"¯ON QUE?":GOTO100
622 PRINT"|IENES UN BONITO PAR DE REMOS!":OB%(31)=CP:OA=1
624 IFBT=1ANDOA=1THENPRINT"♥E ACABO LA MADERA, LO SIENTO!":OB%(40)=0
626 GOTO100
628 IFNO=0THENPRINT"♠CEITAR QUE?":GOTO100
630 GOSUB856
632 IFNO=37THENGOSUB1286:GOTO100
634 PRINT"¯STA TODO GRASIENTO (BEGH!), PERO NO OCURRE NADA.":GOTO100
636 IFNO=0THENPRINT"♠PUNALAR QUE?":GOTO100
638 GOSUB856
640 IFNO=37THENGOSUB1286:GOTO100
642 PRINT"/O ES UNA DE TUS MEJORES IDEAS.¯E TODAS FORMAS, NO TIENES PUNAL."
644 PRINT"(GRACIAS A ¯IOS!)":GOTO100
```

Rutinas de supervivencia

646-664 — *Rutina de ARROJAR. Si el objeto no es el 4 ó el 10 se asume que lo estás dejando caer y el flujo salta a la rutina de dejar. Estas dos acciones son muy similares.*

666-676 — *Rutina de FROTAR. ¿Recuerdas la historia de Aladino y la lámpara maravillosa? Bueno, pues puede que te encuentres con alguna lámpara. Queda saber si llevará o no llevará genio en su interior.*

678-688 — *Rutina de LEER. Puede que haya que leer determinadas inscripciones en rocas, o carteles o notas.*

690-724 — *Rutina de EXAMINAR. Esta es una acción importante que permite conocer más sobre cualquiera de los objetos que aparecen en la aventura. Se puede intentar examinar cualquier cosa. A veces examinar no sólo es interesante, sino absolutamente necesario.*

726-736 — *Rutina de SALTAR. Sólo funciona cuando se está en la posición 29. Hay que tener cuidado con ella, porque puede resultar muy peligrosa.*

```
646 IFNO=0THENPRINT"♠RROJAR QUE?":GOTO100
648 GOSUB856
650 IFOB%(NO)<>-1THENPRINT"|IENES QUE TENERLO PARA ARROJARLO!":GOTO100
652 IFNO=4THEN660
654 IFNO<>10THEN146
656 IFCP<>2THENPRINT"¬L AGUA SE PIERDE EN EL SUELO.":OB%(10)=0:OB%(11)=-1:GOTO10
0
658 PRINT"|AS MOJADO EL AVION.":OB%(10)=0:OB%(11)=-1:WA=1:GOTO100
660 IFCP<>18THENPRINT"¬L BARRO CAE AL SUELO":OB$(21)="LA PALA":OB%(4)=0:GOTO100
662 PRINT"¬L BARRO SE ENDURECE FORMANDO UN CAMINO HASTA LO ALTO DE LA COLINA."
664 P%(18,3)=21:OB%(4)=18:OB$(4)="UN CAMINO DE BARRO SECO.":OB$(21)="LA PALA"
665 MU=1:GOTO100
666 IFNO=0THENPRINT"_ROTAR QUE?":GOTO100
668 GOSUB856
670 IFOB%(NO)<>-1THENPRINT"/O LO LLEVAS ENCIMA.":GOTO100
672 IFNO=37THENGOSUB1286:GOTO100
674 IFNO<>22THENPRINT"ΓOOOOH.....MARAVILLOSO!!":GOTO100
676 PRINT"|U IDEA SOLO HA SERVIDO PARA SACAR BRI- LLO A LA LAMPARA!":GOTO100
678 IFNO=0THENPRINT"LEER QUE?":GOTO100
680 GOSUB856
682 IFNO=51THENPRINT"¯ICE:- /Γ ¬♠¯_ T_ LΓ♥,TL¯♠/Γ♥!":GOTO100
684 IFNO=20THENPRINT"NO SE PUEDEN DESCIFRAR LAS INSCRIPCIONES.":GOTO100
686 IFNO=55THENPRINT"LEE TUS LLAVES.":GOTO100
688 PRINT"/O HAY NADA QUE LEER.":GOTO100
```

```
690 IFNO=0THENPRINT"●UE ES ";N1$;"!":GOTO100
692 GOSUB856
694 IFNO=43ANDCP=2THEN98
696 IFNO=17THENPRINTCO$:OB$(17)="UN GIROCOMPAS":GOTO100
698 IFCP=13AND(NO=45ORNO=46)THENPRINTUP$:GOTO100
700 IFNO=51ORNO=20THEN678
702 IFNO=55THENGOTO678
704 IFNO=52THENPRINTGR$:GOTO100
706 IFNO=47THENPRINT"⁄N CARTEL DICE:-_OMPER EN CASO DE INCENDIO.⁄":BX=1:GOTO100
708 IFNO=37THENGOSUB1286:GOTO100
710 IFNO=27ANDOB%(28)<>-1THENPRINT"⁄AY ALGO EN UN BOLSILLO.":OB%(28)=27:GOTO100
712 IFNO=26THENPRINTDU$:OB%(26)=0:OB%(27)=27:OB%(25)=27:GOTO100
714 IFNO=33THENPRINTRO$:GOTO100
716 IFOB%(25)=-1THENPRINTCN$:CN=1:GOTO100
718 IFNO=58THENPRINTCA$:GOTO100
720 IFNO=40THENPRINT"●E PODRIA HACER UN PUENTE DE TABLAS":OB%(29)=46:GOTO100
722 IFOB%(NO)=-1OROB%(NO)=CPTHENPRINT"⁻S SOLO ";OB$(NO):GOTO100
724 PRINT"⁄O HAY NADA INTERESANTE!":GOTO100
726 IFNO=0THENPRINT"●ALTAR QUE?":GOTO100
728 GOSUB856
730 IFNO<>61ORCP<>29THENPRINT"⁄UEN EJERCICIO, PERO NADA MAS!":GOTO100
732 IFCH=0THENPRINTCH$:CP=30:CH=1:GOTO100
734 IFCH=1THENPRINT"-AES PARA MORIR EN EL ABISMO SIN FONDO!":GOTO1306
736 PRINT"⁄O TIENE SENTIDO CON EL PUENTE AHI DELANTE!": GOTO100
```

738-754 *Rutina de ROMPER. En cierta ocasión tenemos un bote de madera que ya no nos sirve, pero nos hacen falta maderos. Una posible solución puede ser la de romper el bote. Claro que no lo vamos a poder romper con las manos, sino que nos va a hacer falta algún tipo de herramienta.*

756-764 *Rutina de EMPUJAR.*

766-808 *Rutina de SAVE y LOAD. Esta rutina es la que se encarga de almacenar o recuperar el juego en un punto determinado de su desarrollo. Permite utilizar tanto la cinta como el diskette, lo que hace abrir los ficheros correspondientes y leer de ellos o escribir en ellos los valores de todos los contadores, indicadores, variables, posiciones del juego, etc.*

810-828 *Rutina de INSERTAR. En una determinada posición hay que insertar monedas en una ranura. En*

```
738 IFNO=0THENPRINT"_OMPER QUE?":GOTO100
740 GOSUB856
742 IFNO=47ANDBX=1THENPRINT"●HORA NO, SOLO EN CASO DE INCENDIO!!":GOTO100
744 IFOB%(NO)<>-1THENPRINT"⁄O LLEVAS DE ESO!":GOTO100
746 IFOB%(24)<>-1ANDOB%(9)<>-1THENPRINT"-ON QUE?":GOTO100
748 IFNO=30ORNO=3THENOB%(NO)=0:OB%(40)=CP:OB%(8)=CP:OB$(40)="TABLAS CORTADAS":GO
TO754
752 PRINT"-IDICULO!!":GOTO100
754 ZZ=ZZ-1:PRINT"Γ.⁄.":GOTO100
756 IFNO=0THEN690
758 GOSUB856
760 IFNO=37THENGOSUB1286:GOTO100
762 IFNO<>30THENPRINT"⁻SO NO SIRVE DE NADA!":GOTO100
764 PRINT"⁄N BOTE NO SE EMPUJA AL AGUA.  ⁄AY OTRO VERBO.":GOTO100
766 GOSUB78
768 IFDEV=1THENOPEN2,1,1,"PROGRESO":GOTO772
770 OPEN2,8,2,"@0:PROGRESO,S,W"
772 FORI=1TOLO:PRINT#2,OB%(I):NEXTI
774 PRINT#2,RN;C$;CP;C$;PD;C$;ZZ;C$;CH;C$;CL;C$;DO;C$;PA;C$;BO;C$;GR;C$;MU;C$;PH
776 PRINT#2,BT;C$;OA;C$;HE;C$;WA;C$;BX;C$;CN;C$;GL;C$;OB$(19);C$;OB$(6)
778 PRINT#2,OB$(29);C$;OB$(17);C$;OB$(38);C$;OB$(19);C$;OB$(6)
780 PRINT#2,OB$(21);C$;OB$(4);C$;P$(45);C$;OB$(7);C$;OB$(14)
782 PRINT#2,P%(47,1);C$;P%(50,2);C$;P%(28,1);C$;P%(30,0);C$;P%(13,2);C$;P%(21,2)
784 PRINT#2,P%(38,3);C$;P%(2,1);C$;P%(18,3);C$;P%(45,1);C$;P%(48,2)
```

```
786 PRINT#2,NO$(25);C$;OB%(35);C$;P%(30,3);C$;TM;C$;OB$(40)
788 CLOSE2:GOTO98
790 GOSUB78
792 IFDEV=1THENOPEN2,1,0,"PROGRESO":GOTO796
794 OPEN2,8,2,"@0:PROGRESO,S,R"
796 FORI=1TOLO:INPUT#2,OB%(I):NEXTI
798 INPUT#2,RN,CP,PD,ZZ,CH,CL,DO,PA,BO,GR,MU,PH,BT,OA,HE,WA,BX,CN,GL,OB$(19)
800 INPUT#2,OB$(6),OB$(29),OB$(17),OB$(38),OB$(7),OB$(14),OB$(21),OB$(4)
802 INPUT#2,P$(45),P%(29,2),P%(78,0),P%(79,2),P%(47,1),P%(50,2),P%(28,1)
804 INPUT#2,P%(30,0),P%(13,2),P%(21,2),P%(38,3),P%(2,1),P%(18,3)
806 INPUT#2,P%(45,1),P%(48,2),NO$(25),OB%(35),P%(30,3),TM,OB$(40)
808 CLOSE2:GOTO98
810 IFNO=0THEN690
812 GOSUB856
814 IFCP<>45ANDCP<>28THENPRINT"¬ONDE?":GOTO100
816 IFCP=45THEN826
818 IFOB%(25)<>-1THENPRINT"/O TIENES NADA QUE INSERTAR.":GOTO100
820 IFCN<>1THENPRINT"EL MECANISMO SE ATASCA; ESTAS ATRAPADO  PARA SIEMPRE.":GOTO
1306
822 PRINT"¬L CRISTAL SE MUEVE, MOSTRANDO UNA GRAN ROCA.":GL=1
824 IFGL=1THENNO$(25)="MON":OB%(34)=0:OB%(33)=28:GOTO100
826 IFNO<>20THENPRINT"¬AE FUERA; ¬SO NO ES BUENO!":GOTO146
828 PRINTSE$:P%(45,1)=46:P$(45)="DELANTE DE UNA ABERTURA CIRCULAR.":OB%(20)=0:GO
TO100
```







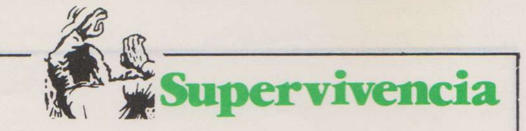
```
830 IFNO=0THEN690
832 GOSUB856
834 IFNO<>30THENPRINT"♠ DONDE? ♠ LA LUNA?":GOTO100
836 IFOB%(30)<>-1THENPRINT"\EJOR COGELO PRIMERO!":GOTO100
838 IFCP<>48ANDCP<>53THENPRINT"♠QUI NO, DESDE LUEGO!":PRINTBT$:GOTO100
840 PRINT"¬L BOTE FLOTA BIEN Y EMPIEZAS A NAVEGAR.":P%(48,2)=50
842 IFOB%(31)<>-1THENPRINT"/O TIENES REMOS Y VAS A LA DERIVA.":GOSUB1290
844 GOTO100
846 IFNO=0THENPRINT"¬ECIR ●/¬?":GOTO100
848 IFNO<>93THENPRINT"¬H?":GOTO100
850 IFCP<>83ANDOB%(39)<>83THENPRINT"¬ARA QUE?":GOTO100
852 PRINT"¬L FENIX SE ASUSTA Y HECHA A VOLAR DEJANDO EL PASO LIBRE!"
854 P%(83,0)=84:OB%(39)=0:PH=1:PRINT"¬STAS DELANTE DEL BOQUETE VACIO.":GOTO100
856 IFNO=6ANDOB%(12)=-1THENNO=12:RETURN
858 IFNO=14AND(OB%(17)=-1ANDOB%(17)=CP)THENNO=70:GOTO872
860 IFNO=16ANDOB%(16)=0THENNO=17
862 IFNO=17AND((OB%(14)=-1OROB%(32)=-1)ANDOB%(15)=-1)THENNO=16:RETURN
864 IFNO=29ANDCP=2THENNO=43:RETURN
866 IFNO=3AND(CP=22ORCP=23)THENNO=41:RETURN
868 IFNO=75ANDCP=45THENNO=58:RETURN
870 IFNO=43ANDCP>4THENNO=29:RETURN
872 IFNO>61THENPRINTGM$:RETURN
874 RETURN
876 P=84:NV=39:NN=93:LO=40:DIMP$(P),P%(P,3),OB$(LO),OB%(NN),VB$(NV),NO$(NN)
```

```
877 C$=CHR$(13)
878 P$(1)="DE PIE AL EXTREMO DE UN GRAN CLARO‾ELANTE TIENES "
880 P$(1)=P$(1)+"UN IMPENETRABLE SETO DE  ARBUSTOS ESPINOSOS.
882 P$(3)=P$(1)
884 P$(4)=P$(1)
886 P$(2)="DE PIE CERCA DE TU AVIONETA, QUE  HA PARADO CERCA DE EL "
888 P$(2)=P$(2)+"EXTREMO DEL CLARO.LA PUNTA DEL ALA
890 P$(2)=P$(2)+" ROZA EL SETO DE ARBUS- TOS."
892 P$(5)="EN LA JUNGLA, AL LADO DEL BOQUETE DEL SETO, TODAVIA HUMEANTE."
894 P$(6)="DE PIE EN LA ORILLA NORTE DE UN"+C$
896 P$(6)=P$(6)+"ARROYO QUE DISCURRE APROXIMADAMENTE AL  NORESTE/SUROESTE"
898 P$(7)="EN LA ORILLA SUR DEL ARROYO."+CHR$(13)
900 P$(7)=P$(7)+" IACIA EL NORTE, ESTE FORMA UNA CHARCA   OSCURA Y PROFUNDA."
902 P$(8)="PERDIDO EN LA JUNGLA."
904 P$(9)=P$(8)
906 P$(10)=P$(8)
908 P$(11)="DE PIE EN UNA ZONA DE LA ORILLA   SUR DEL ARROYO."
910 P$(12)="DE PIE AL EXTREMO DE UN CLARO PER-FECTAMENTE CIRCULAR."
912 P$(12)=P$(12)+C$+"USTO EN EL CENTRO DEL CLARO HAY UN HER-MOSO ARBOL, "
914 P$(12)=P$(12)+"ALTO Y BIEN PROPORCIONADO."
916 P$(13)="EN EL CENTRO DEL CLARO, BAJO EL RAMAJE DEL ƑAS."
918 P$(13)=P$(13)+"•LREDEDOR TIENES EL MISMO SETO DE ARBUSTOS "
920 P$(13)=P$(13)+"ESPINOSOS DE ANTES."
922 P$(14)=P$(8)
```





```
924 P$(15)="DE PIE JUNTO A UNA CHARCA DE BARRO IAY UN FUERTE OLOR A AZUFRE."
926 P$(16)="EN UNA AGRADABLE ZONA DE ARBOLES  DE LA JUNGLA."
928 P$(16)=P$(16)+C$+"¬L AIRE ES BASTANTE FRESCO POR AQUI."
930 P$(17)="EN UNA ZONA EN LA QUE EL SUELO PA-RECE ESTAR MUCHO MAS CALIENTE DE"
932 P$(17)=P$(17)+" LO NOR-MAL.IODAVIA HUELE A AZUFRE."
934 P$(18)="JUNTO A UNA COLINA CON FORMA CONI-CA, RODEADA DE ARBOLES."
936 P$(18)=P$(18)+C$+"¬L TERRENO ES MUY BLANDO Y ABSOLUTAMENTEARIDO."
938 P$(19)="A CIERTA DISTANCIA DE LOS ARBOLES,PERO NO MAS ARRIBA."
940 P$(20)=P$(19)
942 P$(21)="DE PIE JUNTO AL EDIFICIO, QUE NO  PRESENTA NINGUNA ABERTURA."
944 P$(21)=P$(21)+"LA LUZ QUE VISTE PROVIENE DEL REFLEJO DEL SOL EN"
946 P$(21)=P$(21)+" LAS  PULIMENTADAS PAREDES."
948 P$(22)="DE PIE JUNTO A UN PROFUNDO POZO.   ЖA OXIDADA ESCALERA DESCIENDE "
950 P$(22)=P$(22)+"A LAS LO-BREGAS ENTRANAS DE LA TIERRA."
952 P$(23)="DE PIE EN EL FONDO DEL POZO. ЖA DEBIL LUZ LLEGA DE MUY ARRIBA, "
954 P$(23)=P$(23)+"DESDE LA BOCA DEL POZO.
956 P$(24)="EN UNA HABITACION CON UN FUERTE O-LOR A ACEITE."
958 P$(25)="EN LA ZONA NORTE DE UNA CAVERNA;  NO HAY SIGNOS DE NINGUNA FORMA "
960 P$(25)=P$(25)+"DE LUZ NORMAL, PERO CADA DETALLE DE LAS RUGOSAS"
962 P$(25)=P$(25)+"  PAREDES ES ASOMBROSAMENTE VISIBLE."
964 P$(26)="EN EL LADO ESTE DE LA CAVERNA Y ,  PUEDES DISTINGUIR UNA PEQUENA"
966 P$(26)=P$(26)+" CAJA FIJA-DA A LA PARED."
968 P$(27)="EN EL LADO OESTE DE LA CAVERNA."
970 P$(28)="AL EXTREMO SUR DE LA CAVERNA.•OBREUNA RANURA DE LA PARED HAY UN "
```




```
972 P$(28)=P$(28)+"CARTEL QUEDICE■ ■UNAS MONEDAS POR FAVOR■"
974 P$(29)="DE PIE FRENTE A UN GRAN ABISMO. ╭NAESPESA E IMPENETRABLE NIEBLA "
976 P$(29)=P$(29)+"HIERVE EN  LA ORILLA, EN EL SUELO, A TUS PIES."
978 P$(30)="EN EL LADO ESTE DEL ABISMO."
980 P$(31)="EN UN LABERINTO DE ESTRECHOS Y SERPENTEANTES TUNELES."
982 P$(32)=P$(31)
984 P$(33)=P$(31)
986 P$(34)=P$(31)
988 P$(35)=P$(31)
990 P$(36)="AL FINAL DE UN LARGO PASILLO. IACIAEL ESTE LA SALIDA ESTA TAPADA "
992 P$(36)=P$(36)+"POR UN PE-NACHO DE NIEBLA BLANCA QUE SUBE DESDE ELSUELO."
994 P$(37)="EN EL PASILLO; LA NIEBLA LLENA TO-DO EL EXTREMO OESTE."
996 P$(38)="JUSTO ENFRENTE DEL PENACHO DE NIE-BLA, Y HACE MUCHO FRIO."
998 P$(39)="AL FINAL DEL PASILLO. ♠N TRAMO DE  ESCALERAS DE PIEDRA BAJA."
1000 P$(40)="A MEDIO CAMINO EN LOS PELDANOS; ESTOS ESTAN BASTANTE "
1002 P$(40)=P$(40)+"CALIENTES. IAY UNA NO-TA GARABATEADA EN LA PARED."
1004 P$(41)="AL FINAL DE LA ESCALERA.  LAS PAREDES ESTAN MUY CALIENTES."
1006 P$(42)="EN UN PASILLO, CON TUNELES QUE SALEN EN TODAS DIRECCIONES."
1008 P$(43)="EN UN ESTRECHO CONDUCTO.♠ LO LEJOSSE OYE EL "
1010 P$(43)=P$(43)+"CORRER DEL AGUA."
1012 P$(44)="EN UN ESTRECHO CORREDOR.LAS PARE- DES ESTAN MUY CALIENTES, Y "
1014 P$(44)=P$(44)+"EL AGUA QUE  CORRE POR ELLAS SE CONVIERTE EN VAPOR ANTES "
1016 P$(44)=P$(44)+"DE LLEGAR AL SUELO."
1018 P$(48)="DELANTE DE UN GRAN LAGO DE AGUA   HIRVIENTE; A LO LEJOS VES "
```

Rutinas de 'supervivencia

otra posición hay que insertar una piedra, que debemos llevar encima, en la ranura de una pared.

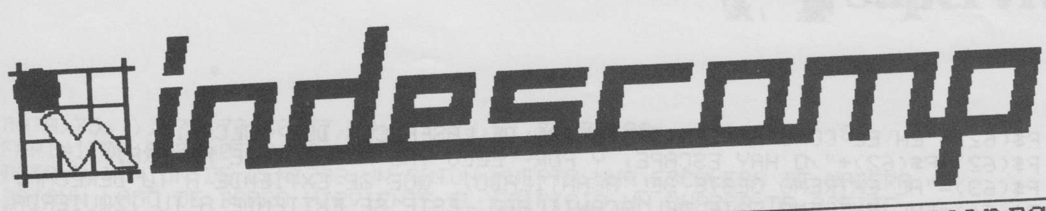
830-844 — *Rutina de LANZAR. Sirve para lanzar un bote al agua.*

846-854 — *Rutina de DECIR. Esta rutina es curiosa. Sirve para decir palabras. En una determinada posición, cerca del final, la única forma de seguir adelante es asustando a un cierto animal. Para asustarle hay que decir «buu», palabreja con la que se le da un susto de muerte.*

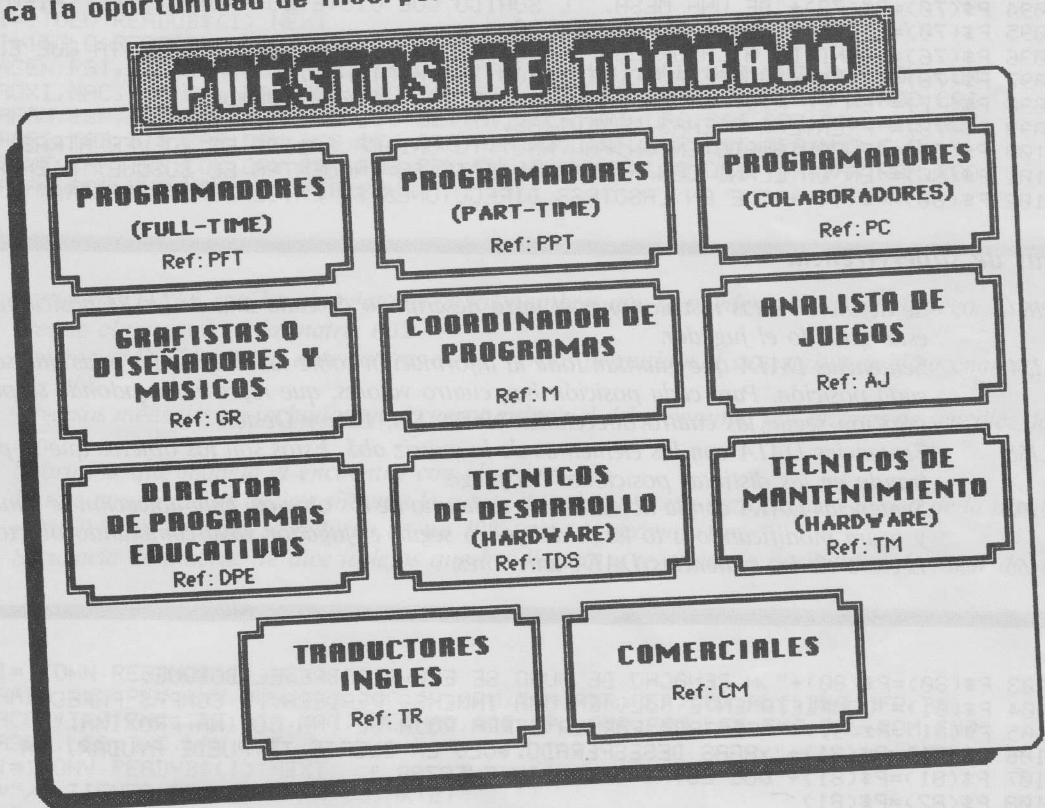
856-874 — *Esta rutina evita repeticiones de nombres como ESCalera y COMpas. Cada vez que tiene que juzgar sobre qué valor asignarle al nombre, lo hace en base a la posición del jugador.*

876 — *Dimensiona todas las matrices del juego. Hay varias matrices fundamentales. Una con los 39 verbos que se pueden utilizar, otra con los 93 nombres permitidos, otra con 40 objetos y una última que contiene las 84 posiciones del juego.*

```
1020 P$(48)=P$(48)+"REBOSAR EL A- GUA A TRAVES DE UNA GRAN VERJA DE HIERRO"
1022 P$(45)="EN UN PASILLO.♠L SUR UNA PARED CONUNA RANURA "
1024 P$(45)=P$(45)+"BLOQUEA EL PASO."
1026 P$(46)="EN UNA ENORME HABITACION-ALMACEN."
1028 P$(47)="EN UNA HABITACION UTILIZADA PARA ALMACEN DE HERRAMIENTAS."
1030 P$(49)="EN UNA HABITACION QUE FUE UTILIZA-DA PARA GUARDAR CHATARRA."
1032 P$(49)=P$(49)+" IAY CAJAS OXIDADAS."
1034 P$(50)="FLOTANDO CERCA DE LA CASCADA DEL  LAGO."
1036 P$(51)="AL ESTE DE LA VERJA; EL AIRE ES HELADOR."
1038 P$(52)="FLOTANDO EN UN ESTRECHO TUNEL; HAYVARIAS SALIDAS."
1040 P$(56)="EN UN BANCO DE ARENA; A LO LEJOS  SE VE UNA DEBIL LUZ Y EL "
1042 P$(56)=P$(56)+"AGUA SE HACE MENOS PROFUNDA HACIA EL NORTE."
1044 P$(57)="DE PIE EN UN SALIENTE SOBRE UNA CAIDA DE UNOS 1000 METROS."
1046 P$(55)="EN TERRENO EMBARRADO.♠ LOS LADOS  HAY ESCARPADAS PAREDES DE ROCA."
1048 P$(53)="APRESURANDOTE HACIA DONDE SE OYE CAER EL AGUA."
1050 P$(54)="EN MEDIO DE LOS RAPIDOS, ♠ TU ALREDEDOR SOLO HAY ROCAS "
1052 P$(54)=P$(54)+"PUNTIAGUDAS QUE PUEDEN HACER ASTILLAS TU FRAGIL BOTE."
1054 P$(58)="AL EXTREMO DEL SALIENTE DE ROCA."
1056 P$(59)=P$(58)
1058 P$(60)="EN LO ALTO DEL ACANTILADO; A AMBOSLADOS EL TERRENO "
1060 P$(60)=P$(60)+"DESCIENDE SUAVEMENTE HACIA EL BOSQUE.‾ETRAS, UN PENACHO "
1062 P$(60)=P$(60)+"DE HUMOSE ELEVA DESDE UNA COLINA PROXIMA."
1064 P$(61)="AL BORDE DE UN VOLCAN EN ACTIVIDAD╱O HAY PELIGRO INMEDIATO, PERO "
1066 P$(61)=P$(61)+"NO SERIA INTELIGENTE SEGUIR ADELANTE."
```



Empresa lider en el sector de INFORMATICA PERSONAL,primer fa-
bricante nacional de periféricos y programas para ordenadores
personales (Commodore,Spectrum,Amstrad,MSX,Spectravideo,etc)
con motivo de la (extraordinaria) expansión prevista para el año
1985,ofrece a todos los jovenes entusiastas de la microinformati-
ca la oportunidad de incorporarse en lo siguientes....



Todos los puestos,ademas de unas excelentes retribuciones y beneficios, contarán con importantes medios para el desarrollo de sus funciones: Equipos de desarrollo,ordenadores, (Sinclair,MSX,Commodore,Apple,Amstrad,Spectravideo,etc.) periféricos, formación profesional,cursos especializados,viajes técnicos,etc.

Es necesario que todos los candidatos posean conocimientos de inglés y de programación,así como un notable interés por su desarrollo profesional en el sector de la Microinformática.

Los interesados deben escribir a :

Avd. Mediterraneo nº9 28007 MADRID

Exponiendo claramente su experiencia y disponibilidad

```
1068 P$(62)="EN EL CONO DE CENIZA DENTRO DE LASFAUCES DEL VOLCAN."
1070 P$(62)=P$(62)+"/O HAY ESCAPE, Y POR  ELLO VAS A MORIR DE HAMBRE."
1072 P$(63)="AL EXTREMO OESTE DEL ACANTILADO,  QUE SE EXTIENDE A TU DERECHA."
1074 P$(64)="AL EXTREMO ESTE DEL ACANTILADO; ESTE SE EXTIENDE A TU IZQUIERDA."
1076 P$(71)=P$(8)
1078 P$(72)=P$(8)
1080 P$(65)="COMPLETAMENTE PERDIDO.♠ LO LEJOS  OYES UN DEBIL 'CHIN-CHIN'."
1082 P$(66)=P$(65)
1084 P$(67)=P$(65)
1086 P$(68)=P$(65)
1088 P$(69)=P$(65)
1090 P$(73)=P$(65)
1091 P$(74)=P$(65)
1092 P$(75)=P$(8)
1093 P$(70)="EN UN CLARO. ⁄N |RIFO DE ORO ESTA SENTADO EN UN LADO, DETRAS "
1094 P$(70)=P$(70)+"DE UNA MESA. ¬L SONIDO QUE OISTE LO PRODUCIA EL |RIFO"
1095 P$(70)=P$(70)+"CONTANDO SU DINERO."
1096 P$(76)="TENDIDO A LA ORILLA DE UN ARROYO. |AS TENIDO SUERTE YA QUE EL "
1097 P$(76)=P$(76)+"GOLPE SOLO  TE DEJO INCONSCIENTE."
1098 P$(77)="EN EL RIO, EN UNA AMPLIA RIBERA O PLAYA DORADA."
1099 P$(78)="FRENTE A ARENAS MOVEDIZAS."
1100 P$(79)="ENPANTANADO EN UN PANTANO; TU UNI-CA SALIDA ES VOLVER ATRAS."
1101 P$(80)="EN UN LLANO CON HIERBA.♠L ESTE SE ENCUENTRA EL BOSQUE, MIENTRAS "
1102 P$(80)=P$(80)+"QUE EN LASOTRAS DIRECCIONES SE EXTIENDE LA PRADERA"
```

Rutinas de supervivencia

877-1.116 — *Se define en estas instrucciones el texto descriptivo de cada una de las 84 posiciones en que puede estar situado el jugador.*

1.118-1.134 — *Sentencias DATA que guardan toda la información sobre las direcciones a las que se puede ir desde cada posición. Para cada posición hay cuatro valores, que representan adonde se puede ir desde la posición según las cuatro direcciones Norte, Sur, Este y Oeste.*

1.138-1.158 — *Sentencias DATA con los elementos de la matriz ab$. Estos son los objetos que se pueden ir encontrando en las distintas posiciones del juego.*

1.160-1.162 — *Sentencias DATA con la ubicación de cada uno de los objetos. Esta ubicación se almacenará en ob% y se irá modificando a lo largo del juego según el jugador vaya cambiando objetos de sitio.*

1.164-1.166 — *Lectura de las sentencias DATA anteriores.*

```
1103 P$(80)=P$(80)+" ⁄N PENACHO DE HUMO SE ELEVA SOBRE EL BOSQUE."
1104 P$(81)="COMPLETAMENTE SOLO EN UNA INMENSA PRADERA.|U COMPAS PARECE "
1105 P$(81)=P$(81)+"AFECTADO POR LATIERRA ROJA DE UNA COLINA PROXIMA."
1106 P$(81)=P$(81)+"XAGAS DESESPERADO.♠OLO LA SUERTE TE PUEDE AYUDAR, YA "
1107 P$(81)=P$(81)+"QUE ESTAS CASI SIN FUERZAS."
1108 P$(82)=P$(81)
1109 P$(83)="DE NUEVO EN LA JUNGLA, MILAGROSA- MENTE DELANTE DEL BOQUETE"
1110 P$(83)=P$(83)+" QUE HICISTE ALQUEMAR EL SETO."
1112 P$(84)="DE NUEVO EN EL CLARO DONDE CAYO TU AVIONETA."
1113 P$(84)=P$(84)+C$+"¬STAMOS SEGUROS DE QUE HABRAS APRENDIDO ALGO, Y "
1114 P$(84)=P$(84)+"QUIZAS LA PROXIMA VEZ SEAS MAS  CUIDADOSO."
1115 P$(84)=P$(84)+C$+"¬E CUALQUIER FORMA AQUI ESTA EL HELICOP-TERO."
1116 P$(84)=P$(84)+"!_ELICIDADES Y HASTA LA PROXIMA!"
1118 DATA,2,4,3,1,,4,3,1,2,4,,1,2,,3,2,6,,,5,11,7,8,,10,12,6,,9,6,
1119 DATA8,,11,,7,11,12,11,6,,10,9,7,14,13,10,,,,12,10,,12,,13,17,,16
1120 DATA,18,15,,15,,,,19,20,16,,,,18,20,,,18,19,19,20,18,22,,
1121 DATA,,,,24,,,23,25,24,28,26,27,25,28,,27,25,28,26,,25,,26,27
1122 DATA,,28,,31,,,30,31,32,31,33,31,31,31,31,31,31,34,31,35,31,31
1123 DATA31,31,31,36,,,35,37,,,36,38,,,37,,,,38,40,,,39,41,,,40,42
1124 DATA43,45,41,,44,42,,,48,43,,,42,,,,45,,47,,,,,46,,44,,
1125 DATA47,,,,,,,48,,,52,50,53,55,56,51,54,52,,,,,,,52,,,
1126 DATA57,52,,,,56,59,58,,,57,,,,,57,,61,64,63,60,62,,,,,,
1128 DATA65,,60,71,,,75,60,66,63,65,65,73,65,65,72,65,68,65,65,65,65,69,65
1130 DATA70,65,65,65,,69,,,,,63,,,,66,,,66,74,,,67,,73,,,,64
```

```
1132 DATA,75,,,,,78,70,80,77,79,,,,,78,81,78,83,82,,,,,,,,,,6,,80,,,
1134 FORI=1TOP:FORJ=0TO3:READP%(I,J):NEXTJ,I
1136 DATAUN MONTON DE CENIZAS,UN RATON MUERTO,UNA ESCALERA DE MADERA
1138 DATAUN LAGO DE BARRO HIRVIENTE,EL EDIFICIO,UN CHARCO DE ACEITE
1140 DATAUNA VERJA CON PESTILLO Y CANDADO,UNOS CLAVOS OXIDADOS,UN MARTILLO
1142 DATAUNA LATA DE AGUA,UNA LATA VACIA,UNA LATA DE ACEITE,UNA CAJA DE CERILLAS
1144 DATAUN ROLLO DE CUERDA,UN IMAN,UN COMPAS HECHO A MANO,UN COMPAS
1146 DATAUNA BOTELLA DE OXIGENO,UN MACHETE OXIDADO,UNA PIEDRA CON GRABADOS
1148 DATAUNA VIEJA PALA,UNA VIEJA LAMPARA,UNA ANTORCHA ENCENDIDA, UN HACHA
1150 DATAVARIAS MONEDAS,UN ESQUELETO MEDIO VESTIDO,UN FARDO DE ROPAS
1152 DATAUN MANOJO DE LLAVES,VARIOS MADEROS,UN BOTE,UN PAR DE REMOS,UNA LAZADA
1154 DATAUNA GRAN ROCA,UNA GRUESA LAMINA DE CRISTAL,UN TEMIBLE FANTASMA
1156 DATAUN ESPEJITO,UN BASILISCO DE BRONCE,UNA PUERTA CERRADA,UN FENIX DE PLATA
1158 DATAMADEROS SIN CORTAR
1160 DATA,13,,15,21,24,50,49,26,2,,,-1,2,2,,2,2,8,40,18,22,,49,,27,,,,,,,,28
1162 DATA38,14,13,47,84,46
1164 FORI=1TOLO:READOB$(I):NEXT
1166 FORI=1TOLO:READOB%(I):NEXT
1168 DATACEN,RAT,ESC,BAR,EDI,ACE,VER,CLA,MAR,AGU,LAT,ACE,CER,CUE,IMA,COM,COM
1170 DATAOXI,MAC,PIE,PAL,LAM,ANT,HAC,MON,ESQ,ROP,LLA,MAD,BOT,REM,LAZ,ROC,CRI
1172 DATAFAN,ESP,BAS,PUE,FEN,MAD,ESC,SET,AVI,ALA,ARB,UPA,CAJ,POZ,RAN,NIE,NOT
1174 DATAGRI,TAB,DIN,CAR,CAN,PES,DEN,PRO,ABA,ABI,NOR,SUR,EST,OES,N,S,E,O
1176 DATACUE,CHA,GRO,COL,CAV,PAR,PAS,PEL,CUL,SLU,HAB,LAG,CAS,TUN,BAN,RAP,ROC
1178 DATASAL,ACA,HUM,VOL,RIO,PLA,BUU
```

l.168-l.180	*Sentencias DATA con las abreviaturas de los nombres que se pueden utilizar en el juego. Estos nombres se almacenan en la matriz no$.*
l.182-l.188	*Sentencias DATA con las abreviaturas de los verbos permitidos. Estos verbos se almacenan en la matriz vb$.*
l.190-l.284	*Diversos mensajes no incluidos en el cuerpo principal del programa, por razones de sencillez de escritura.*
l.286-l.288	*Subrutina que maneja el encuentro con el objeto 37.*
l.290-l.304	*Rutina encargada de decirte si tienes la suerte a tu favor. Cada vez que sales airoso de la prueba, tus posibilidades de éxito se reducen en un 10% para la próxima vez.*
l.306-l.320	*Secuencia de muerte. Te dice lo lejos que has llegado y te ofrece la posibilidad de jugar de nuevo.*

```
1180 FORI=1TONN:READNO$(I):NEXT
1182 DATAABA,AYU,REV,INV,PUN,DEJ,COG,IR,CRU,TOM,ABR,CER,COM,ALI,BEB,OFR,BAL
1184 DATACOR,TRO,ENC,TRE,ATA,MAT,GOL,HAC,ACE,APU,ARR,FRO,LEE,EXA,SAL,ROM,EMP
1186 DATASAV,LOA,INS,LAN,DEC.
1188 FORI=1TONV:READVB$(I):NEXT
1190 DI$="/O TIENES COMPAS PARA GUIARTE!"
1192 SC$="XAYA TONTERIA, ES PARTE DEL ESCENARIO!"
1194 HE$="¯S DEMASIADO PESO PARA TUS POCAS FUERZAS."
1196 AT$="¯STA UNIDO A ALGO MAS."
1198 HA$="/O ES LO BASTANTE SOLIDO COMO PARA COGERLO Y SE ESCAPA DE LOS DEDOS."
1200 CO$="¬L GIROCOMPAS ES PARTE DEL PANEL DE MAN-DOS, Y NO SE PUEDE COGER."
1202 BU$="|E QUEMAS LA MANO, PERO NO OCURRE NADA MAS."
1204 DU$="♠E DESHACE EN POLVO."
1206 ST$="♠E.VUELVE Y TE MIRA A LOS OJOS."
1208 SO$="¬UEDES QUEDAR PETRIFICADO."
1210 GM$="/O SE MUY BIEN QUE INTENTAS, ¯♠T¯┘┌ QUE SOLO QUIERAS SER GRACIOSO!"
1212 GN$="¬L |RIFO, CONTRARIAMENTE A LO QUE SE PIENSA ES "
1214 GN$=GN$+"INTOCABLE, SIN EMBARGO ACEPTA O- FRECIMIENTOS.!"
1216 AC$="¬L |RIFO ACEPTA CON INCREIBLE RAPIDEZ, YTE MUESTRA UN CAMINO AL NORTE"
1218 BA$="¬L |ASILISCO ES UNA CRIATURA DORMILONA, PERO SU SENTIDO DEL TACTO "
1220 BA$=BA$+"Y SU OIDO SON MUY AGUDOS,"+C$+"_ECUERDA A \EDUSA!"
1222 UP$="¬L /PAS ES UN ARBOL LEGENDARIO, DEL QUE SE DICE QUE ENVENENA EL AIRE "
1224 UP$=UP$+"A SU ALREDEDOR Y QUE QUIEN RESPIRE SUS VAPORES, MO-RIRA!"
1226 GR$="LA LEGENDARIA AGUDEZA AUDITIVA DEL GRIFOHA SIDO HECHADA POR TIERRA;"
```

```
1228 GR$=GR$+" ESTE ESTA MASSORDO QUE UNA TAFIA!"
1230 RO$="LAS PALABRAS:-"+CHR$(13)+" ▩▩⁄⁄▁▔|♠|▔ ▔▔ ||♠⁄▁⁄♠⁄|▔-⁄♠⁄|-⁄▔♠⁄"
1232 RO$=RO$+C$+"RODEAN LA ROCA, QUE HUELE A MENTA.!"
1234 RO$=RO$+C$+"(|E LO DIJE, ES UNA GRAN ROCA)"
1236 CN$=" |AY DOS MONEDAS DE ORO, UNA DE PLATA, Y "
1238 CN$=CN$+"OTRA DE HIERRO."
1240 BR$="LAS TABLAS ATRAVIESAN EL ABISMO, UNIENDOSE FUERTEMENTE."
1242 CH$="▔STAS DE PIE EN EL SUELO DEL ABISMO, TU PIE OCULTO POR LA NIEBLA Y "
1244 CH$=CH$+"TUS OJOS A LAALTURA DEL SUELO DE LA CAMARA."
1246 CH$=CH$+C$+"♠ MEDIDA QUE TE LEVANTAS, VES ENTRE EL POLVO, QUE "
1248 CH$=CH$+"DELANTE TIENES UN NUEVO E IN-SONDABLE ABISMO."
1250 BO$="▔L PESTILLO SALTA CONVERTIDO EN TROCITOSDE OXIDO."
1252 GA$="LA VERJA SE DOBLA BAJO LA FUERZA DEL A- GUA."
1254 MI$="▔L ESPEJO REFLEJA LA MIRADA DEL BASILIS-CO"
1256 PD$="▔STA MUY OSCURO; SI SIGUES ADELANTE PUE-DES CAER EN UN POZO.
1258 FH$=" ⁄| ▁ENIX HA NACIDO DE LAS CENIZAS DEL SETO Y BLOQUEA EL PASO."
1260 EX$="LAS LLAMAS PRENDEN EL ALA DEL AVION Y SEPRODUCE UNA EXPLOSION QUE "
1262 EX$=EX$+"PUEDE COSTARTELA VIDA."
1264 EX$=EX$+C$+"(▔ARA QUE CREES QUE SIRVE EL AGUA?)"
1266 SE$="LA PIEDRA Y LA RANURA ENCAJAN PERFECTA- MENTE; LA PARED GIRA "
1268 SE$=SE$+"REVELANDO UN CAMINOHACIA DELANTE."
1270 OI$="▔L ACEITE EMPAPA EL TERRENO DEJANDO SOLAMENTE UNA MANCHA."
1272 OL$="▔L ACEITE SE EXTIENDE Y FORMA UN CHARCO."
1274 MR$="▔L ESPEJO SE ROMPE EN MIL PEDACITOS."
1276 FA$="▔STA LEJOS Y EL TERRENO ES MUY BLANDO;NECESITARIAS UN LAZO DE VAQUERO.
```





```
1278 PL$="LAS CENIZAS DEL AVION SE ESPARCEN POR ELAIRE."
1280 CA$="▔N LA BASE DE UNA DEPRESION HAY UNAS CU-RIOSAS INSCRIPCIONES."
1282 JA$="▔STAN UNIDOS! ▔ONCENTRATE!"
1284 RETURN
1286 IFOB%(36)<>-1THENPRINTST$:PRINTSO$:GOSUB1290:RETURN
1288 IFOB%(36)=-1THENPRINTST$:PRINTMI$:RETURN
1290 PRINT"▩▔|▁♠♠ ⁄♠ |▔|▁♠ ▔♠-♠ ▔▔⁄▔-- | ⁄ ♠▔▁|▔|"
1292 GETA$:IFA$=""THEN1292
1294 IFINT(RND(0.5)*RN)+1<10THEN1306
1296 PRINT"▩|UENO, POR ESTA VEZ LO HAS CONSEGUIDO.  ♠IN EMBARGO NO ESPERES ";
1298 PRINT"TANTA CLEMENCIA  PARA LA PROXIMA VEZ!"
1300 RN=RN-10
1302 PRINT"|US POSIBILIDADES DE SOBREVIVIR LA PROXIMA VEZ SON DE "RN/10" A 1!"
1304 FORF=1TO5000:NEXT:RETURN
1306 FORF=1TO5000:NEXT:POKE53280,0:POKE53281,11
1308 PRINT"▩▩▩|AS MUERTO, TU CUERPO SE HA DISUELTO EN EL ESCENARIO, NADIE TE LL
ORA."
1310 PRINT"▩▩(♠UE TRISTE, NO ES VERDAD?)"
1311 PRINT"▩▩|AS COMPLETADO EL...▩":PRINTINT(CP*100/84)"▩% DEL TRAYECTO TOTAL."
1312 PRINT"▩▩♠UIERES INTENTARLO OTRA VEZ(♠⁄⁄)?"
1314 GETPA$:IFPA$=""THEN1314
1316 IFPA$="S"THENRUN
1318 IFPA$="N"THENPRINT"♠DIOS!.":POKE53280,14:POKE53281,6:PRINT"▩":END
1320 GOTO1312
```

MODIFICACIONES A LA PRIMERA PARTE DE SUPERVIVENCIA

Errare humanum est, y como en Commodore Magazine todos somos humanos, (incluso los micros que tenemos tienen su poquito de humanidad) cometemos errores de cuando en cuando. El caso es que el listado que publicamos de la primera parte de Supervivencia no era la versión definitiva, como era nuestra intención publicar, sino que (seguro que algún duendecillo anda de por medio) publicamos una de las versiones de prueba y de corrección de errores, y claro hay una serie de sentencias o instrucciones incompletas o incorrectas. El listado que damos a continuación incluye todas estas instrucciones, pero ahora, esta vez sí, totalmente corregidas. Lo único que hay que hacer es reemplazar, en el listado del mes anterior, las instrucciones incorrectas por estas que adjuntamos. Solamente hay una instrucción nueva, la 211. El resto estaban incluidas en el listado anterior.

También tenemos que rectificar la descripción de ciertas rutinas, que quedan como sigue:

146-178	*Rutina de DEJAR. (Anteriormente, el mes pasado, habíamos dicho «soltar» en lugar de «dejar»)*
180-230	*Rutina de COGER*
232-258	*Rutina de IR*

```
UN MONTON DE CENIZAS                 UNA VIEJA PALA
UN RATON MUERTO                      UNA VIEJA LAMPARA
UNA ESCALERA DE MADERA               UNA ANTORCHA ENCENDIDA
UN LAGO DE BARRO HIRVIENTE           UN HACHA
EL EDIFICIO                          VARIAS MONEDAS
UN CHARCO DE ACEITE                  UN ESQUELETO MEDIO VESTIDO
UNA VERJA CON PESTILLO Y CANDADO     UN FARDO DE ROPAS
UNOS CLAVOS OXIDADOS                 UN MANOJO DE LLAVES
UN MARTILLO                          VARIOS MADEROS
UNA LATA DE AGUA                     UN BOTE
UNA LATA VACIA                       UN PAR DE REMOS
UNA LATA DE ACEITE                   UNA LAZADA
UNA CAJA DE CERILLAS                 UNA GRAN ROCA
UN ROLLO DE CUERDA                   UNA GRUESA LAMINA DE CRISTAL
UN IMAN                              UN TEMIBLE FANTASMA
UN COMPAS HECHO A MANO               UN ESPEJITO
UN COMPAS                            UN BASILISCO DE BRONCE
UNA BOTELLA DE OXIGENO               UNA PUERTA CERRADA
UN MACHETE OXIDADO                   UN FENIX DE PLATA
UNA PIEDRA CON GRABADOS              MADEROS SIN CORTAR
```



```
8 XZ=FRE(0):PRINT:PRINT"■■ Γ_♠ QUE?■";:GOSUB30:PRINT"■■"
64 IFCP=54THENGOSUB1290:CP=76:OB%(30)=0:OB$(40)="MADERA CORTADA":OB%(40)=CP:GOTO
98
150 IFOB%(NO)<>-1THENPRINT"|IENES QUE TENERLO PARA DEJARLO.":GOTO100
198 IFNO=40ANDTM=0THENPRINT"PRIMERO HAY QUE CORTAR LOS MADEROS.":GOTO100
204 IFNO=58ORNO=5THEN PRINT"|IRAR JUNTOS":GOTO100
209 IFNO=4ANDOB%(21)=-1THENOB$(21)="UNA PALA CON BARRO":OB%(4)=-1:PRINT"Γ.⁄."
210 IFNO=4ANDOB%(21)=-1THENOB$(4)="":GOTO100
211 IFNO=6ANDOB%(11)<>-1THENPRINT"¬N QUE?":GOTO100
228 ZZ=ZZ+1:IFZZ>7THENPRINT"/O PUEDES LLEVAR NADA MAS!":ZZ=ZZ-1:GOTO100
248 IF(CP=13ANDNO=2)ANDHE<>2THENPRINT"|ODAVIA!":GOTO100
294 IFPA<>1ANDBO<>1THENPRINT"LA VERJA TIENE PESTILLO Y CANDADO"
296 IFPA<>1ANDBO=1THENPRINT"LA VERJA TIENE CANDADO PERO NO PESTILLO"
298 IFPA=1ANDBO<>1THENPRINT"LA VERJA TIENE PESTILLO PERO NO CANDADO"
312 IFNO=33THENPRINT"LA ROCA NO SE MUEVE,PERO TUS MANOS SABENA MENTA.":GOTO100
314 IFNO=34THENPRINT"¬L CRISTAL PARECE MUY SOLIDO.":GOTO100
394 IFNO=40THENPRINT"LOS MADEROS ESTAN CORTADOS":OB$(40)="MADEROS CORTADOS":TM=1
:GOTO100
398 IFCL<>2THENPRINT"/O TIENE OBJETO SIN SABER ANTES DONDE VAS.":GOTO100
400 PRINT"¬L MACHETE CORTA EL SETO MOSTRANDO UN CAMINO AL ESTE":HE=2
402 PRINT"♦IN EMBARGO, CON EL ULTIMO GOLPE, EL MA-CHETE SE ROMPE EN TROZOS."
```

Concurso

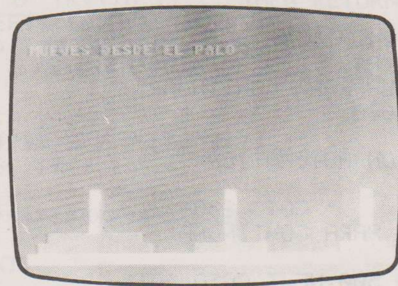
Las torres de Hanoi

CBM 64

Hay determinados programas que podríamos llamar clásicos y de los que se pueden encontrar numerosas versiones, con pequeñas variantes, que aparecen de cuando en cuando. Un ejemplo de ello lo tenemos en este programa sobre el juego de las torres de Hanoi (o torres de Brahama).

La versión que ahora os presentamos nos la ha enviado Antonio Lozano desde Barcelona y está diseñada para funcionar en un Commodore 64. Nosotros publicamos una versión de este conocido juego en el n.º 1 de Commodore Magazine, pero la versión que nos envía Antonio supone una agradable mejora.

El programa incluye las instrucciones para realizar los movimientos, así como las reglas generales del juego. Inicialmente ofrece la posibilidad de elegir el número de discos con que se va a jugar (entre 1 y 6). Otra característica de esta versión es que permite, pulsando una tecla, obtener la solución de forma automática y con el mínimo número de movimientos. Esta forma automática de resolver el problema se puede llevar a cabo de dos maneras distintas, que sólo difieren en la forma de llevar a cabo las pausas entre los movimientos de los anillos. Por último, también está incluida la posibilidad de cambiar en cualquier momento el número de anillos con que se juega.



El programa está estructurado como reseñamos a continuación.

5-15	Inicialización.
18-30	Entrada del número de discos.
35-176	Definición de palabras gráficas y dibujo inicial.
180-260	Selección de opciones y salto a subrutina adecuada.
999-1090	Solución manual.
1999-2100	Solución automática.
3999-6110	Dibujo de posición inicial y subrutina de instrucciones.
9999-20030	Subrutinas de movimiento de discos y sonido.

PREMIADO CON 5.000 PESETAS

```
5 FOR L=54272 TO 54295:POKE L,0:NEXT
6 POKE 54296,8:POKE 54277,9:POKE 54278,0:POKE 54273,17:POKE 54272,37
10 POKE 53280,11:POKE 53281,11
15 GOSUB 6000
18 CLR
19 DEF FN SIG(X)=X+1-INT(X/3)*3
20 PRINT "▣CUANTOS DISCOS (1-6) ?"
25 GET D$:IF D$="" THEN 25
26 D=VAL(D$)
30 IF D>6 OR D<1 THEN GOTO 20
35 S$="                "
50 B$="██████████████████████████"
70 I$="███████████"
100 PRINT "▣□";:FOR I=1 TO 4:PRINT"                                        ";:NE
XT
150 T$(1)="▪██████"+LEFT$(B$,22-D)
160 T$(2)="▪█████████████████████"+LEFT$(B$,22-D)
170 T$(3)="▪████████████████████████████████████"+LEFT$(B$,22-D)
172 DIM TR(3,D)
173 GOSUB 4000
174 FOR I=1 TO D:TR(1,I)=I:NEXT
176 H(1)=1:H(2)=D+1:H(3)=D+1
180 PRINT"▣OPCION:                 "
190 PRINT"F1: MANUAL"
200 PRINT"F3: AUTOMATICO CON PAUSAS MANUALES"
210 PRINT"F5: AUTOMATICO CON PAUSAS AUTOMATICAS"
217 PRINT"F7: CAMBIAR NUMERO DE DISCOS"
```

Concurso

```
220 GET O$:IF O$="" THEN 220
230 IF O$="■" THEN GOSUB 1000:GOTO 173
240 IF O$="■" THEN P=0:GOSUB 2000:GOTO 173
250 IF O$="■" THEN P=1:GOSUB 2000:GOTO 173
255 IF O$="■" THEN GOTO 18
260 GOTO 220
999 REM ***************** SOLUCION MANUAL
1000 GOSUB 5000:M=0:TI$="000000"
1001 PRINT"■■■■■MUEVES DESDE EL PALO";
1010 GET X$:IF X$="" THEN GOSUB 7000:GOTO 1010
1020 PRINT"■■■■■HASTA EL      "
1030 GET Y$:IF Y$="" THEN GOSUB 7000:GOTO 1030
1035 X=VAL(X$):Y=VAL(Y$)
1036 IF X<1 OR X>3 OR Y<1 OR Y>3 THEN GOTO 1001
1040 GOSUB 10000:M=M+1
1050 IF H(3)<>1 THEN 1001
1055 R$=TI$
1060 PRINT"■■          F E L I C I D A D E S      "
1062 PRINT "■LO HAS HECHO EN";M;"MOVIMIENTO";
1063 IF M<>1 THEN PRINT "S"
1064 IF M=1 THEN PRINT
1065 PRINT"[ EL MINIMO ES";2↑D-1;"]"
1067 PRINT"■ TIEMPO: ";MID$(R$,3,2);":";MID$(R$,5,2)
1070 PRINT"■■■PULSA UNA TECLA PARA CONTINUAR"
1080 GET K$:IF K$="" THEN 1080
1090 RETURN
1999 REM ************ SOLUCION AUTOMATICA
2000 RE=1
2004 IF P=1 THEN PRINT"■                 ":GOTO 2006
2005 PRINT"■PULSA UNA TECLA PARA CONTINUAR"
2006 GOSUB 5000
2007 IF P=1 THEN FOR I=1 TO 500:NEXT:GOTO 2010
2008 GET K$:IF K$="" THEN 2007
2010 X=RE:Y=FN SIG(RE):RE=FN SIG(RE)
2015 IF INT(D/2)<>D/2 THEN Y=FN SIG(Y):RE=FN SIG(RE)
2017 GOSUB 10000
2020 IFH(3)=1THENPRINT"■                              ":FORI=1TO3000:NEXT:RETURN
2025 IF P=1 THEN FOR I=1 TO 500:NEXT:GOTO 2040
2030 GET K$:IF K$="" THEN 2030
2040 X=FN SIG(Y):Y=FN SIG(X)
2045 IF H(X)>=D+1 THEN TEM=X:X=Y:Y=TEM:GOTO 2070
2050 IF H(Y)<D+1 THEN IF TR(X,H(X))>TR(Y,H(Y)) THEN TEM=X:X=Y:Y=TEM
2070 GOSUB 10000
2100 GOTO 2007
3999 REM * DIBUJO DE LA POSICION INICIAL
4000 PRINT"■";LEFT$(B$,23);"■■                              ";
4010 FOR I=22 TO 22-D STEP -1
4020 PRINT "■■■■■■■";LEFT$(B$,I);"■■ ";
4030 PRINT SPC(12);"■■ ";
4040 PRINT SPC(12);"■■";
4050 NEXT
4060 FOR I=D TO 1 STEP -1
4070 PRINT "■■■■■■■■■";LEFT$(B$,22-D+I);LEFT$(I$,I);LEFT$(S$,I*2+1)
4080 NEXT
4090 RETURN
5000 PRINT"■";:FOR I=1 TO 5:PRINT"                              ";:NEX
T
5010 RETURN
6000 INPUT "■■QUIERES INFORMACION ";I$
6010 IF MID$(I$,1,1)="N" THEN RETURN
6020 IF MID$(I$,1,1)<>"S" THEN GOTO 6000
```

```
6030 PRINT"█EN LA PANTALLA APARECERAN TRES PALOS.█"
6040 PRINT"█EL JUEGO CONSISTE EN TRASLADAR LOS          DISCOS DEL PALO 1";
6050 PRINT" AL PALO 3 SIGUIENDO   LAS SIGUIENTES REGLAS:█"
6060 PRINT"███ HAY QUE TRASLADAR LOS DISCOS DE UNO        EN UNO"
6070 PRINT"███ NO SE PUEDE COLOCAR UN DISCO SOBRE        OTRO DE MENOR RADIO"
6075 PRINT"███ LOS DISCOS SE TOMAN Y SE DEPOSITAN "
6076 PRINT"  EN LA PARTE SUPERIOR DE LOS PALOS"
6080 PRINT"███(LA DENOMINACION DE LOS PALOS ES 1,2 Y 3 DE IZQUIERDA A DERECHA)"
6084 PRINT"█INTENTA HACERLO EN EL MENOR NUMERO DE   MOVIMIENTOS"
6090 PRINT"██PULSA UNA TECLA PARA COMENZAR█"
6100 GET K$:IF K$="" THEN GOTO 6100
6110 RETURN
7000 PRINT"█";MID$(TI$,3,2);":";MID$(TI$,5,2)
7020 RETURN
9999 REM ***************** MOVER UN DISCO
10000 IF H(X)=D+1 THEN RETURN
10010 T=TR(X,H(X))
10012 IF H(X)=D+1 THEN RETURN
10015 T=TR(X,H(X))
10017 IF H(Y)<D+1 THEN IF TR(Y,H(Y))<T THEN RETURN
10020 PRINT"█";T$(X);LEFT$(B$,H(X));LEFT$(I$,T);
10030 PRINT LEFT$(S$,T);"██ ██";LEFT$(S$,T)
10035 GOSUB 20000
10040 H(X)=H(X)+1
10045 H(Y)=H(Y)-1:TR(Y,H(Y))=T
10050 PRINT T$(Y);LEFT$(B$,H(Y));LEFT$(I$,T);
10060 PRINT "█";LEFT$(S$,T*2+1)
10070 RETURN
19999 REM ************************ SONIDO
20000 POKE 54276,33
20010 FOR S=1 TO 70:NEXT
20020 POKE 54276,32
20030 RETURN
```

PROGRAMA COMPROBADO commodore Magazine

Concurso

Biorritmos

VIC-20

Ramón Marín, lector de nuestra revista en Bilbao, ha enviado a concurso este programa para el VIC-20 (sin expansiones de memoria) titulado BIORRITMOS. El programa calcula y representa los biorritmos de cualquier persona que introduzca los datos sobre su fecha de nacimiento y la fecha para la que desea conocer el biorritmo. Las fechas se introducen respondiendo a las preguntas que formula el ordenador y teniendo en cuenta que los meses hay que escribirlos con letras y los años con cuatro cifras. Los biorritmos pueden aparecer representados de dos formas distintas. Un primer modo es en alta resolución, en el que se puede ver el valor de los biorritmos para un mes completo, día por día. El orden en el que aparecen los tres biorritmos es el inverso al alfabético, es decir, primero aparecerá representado el biorritmo intelectual, a continuación el físico y por último el emocional.

El otro modo de representación es en baja resolución. En este caso aparecen representados únicamente veintiún días, pero con la ventaja de que cada uno de los biorritmos lleva asociada una letra (la inicial de su nombre), con lo que resulta más sencillo saber cual es cual.

Para los que no lo recuerdan diremos que valores de una curva por encima del eje y cerca del máximo significan condiciones propicias (ya sean intelectuales, físicas o emocionales), mientras que valores por debajo del eje y cerca del mínimo significan condiciones desfavorables. Los días en los que las curvas cortan al eje representan días inestables, de cambio. Por último, los días en los que las tres curvas cortan al eje simultáneamente son días críticos, ¡cuidadito con ellos!

Para los que quieran hacer modificaciones o simplemente entender el funcionamiento del programa adjuntamos unas líneas con la estructura.

20-110	Entrada de datos y cálculo de los días vividos.
225-240	Petición de resolución
250-267	Representación del biorritmo intelectual.
269-280	Idem para el físico.
281-286	Idem para el emocional.
290-292	Representación de la barra para los días críticos.
350-480	Meses, días de los meses y cálculo de años bisiestos.
570-1000	Final.




```
0 REM***BIORRITMOS***  1REM** RAMON MARIN**
2 REM**NOVIEMBRE 84**
5 PRINT"HOLA SOY TU ORDENADOR       PERSONAL"
11 PRINT"ESTE PROGRAMA CALCULA TUS BIORRITMOS"
12 PRINT"DESDE LA FECHA ACTUAL HASTA PASADO UN MES":PRINT"INTRODUCE LOS A O
S"
13 PRINT"CON CUATRO CIFRAS":FORL=0TO9000:NEXT
20 PRINT"FECHA DE TU NACIMIENTO":INPUT"DIA";D
30 INPUT"MES";M$:INPUT"A O";Y:DIMX(12):PRINT"":B=Y:GOSUB350:FORI=1TO12
40 READX(I):NEXTI:GOSUB475:IFM<12THEN43
41 R=X(M)-D:GOTO50
43 R=X(M)-D:FORL=MTO11:M=M+1:R=R+X(M):NEXTL
50 PRINT"      FECHA ACTUAL":INPUT"DIA";DD:INPUT"MES";M$:INPUT"A O";A:PRINT"
"
70 GOSUB350:FORL=1TO12:READX(L):NEXTL:Y=A:GOSUB475:IFM<12THEN90
75 DATA31,28,31,30,31,30,31,31,30,31,30,31
80 Q=X(M)-DD:GOTO100
90 Q=X(M)-DD:FORL=MTO11:M=M+1:Q=Q+X(M):NEXTL
100 SS=R:FORKK=BTOA:SS=SS+W:W=365:IFKK/4=INT(KK/4)THENW=366
110 NEXTKK:SS=SS-Q+1:F=SS:RR=SS
225 PRINT"INDICA LA RESOLUCION  ALTA=A                    BAJA=B"
230 GETA$:IFA$="B"THEN490
240 IFA$<>"A"THEN230
250 POKE36869,255:FORI=7168TO7679:POKEI,0:NEXT:POKE36879,8:PRINTCHR$(147)
260 FORL=0TO7:FORM=0TO7:POKE7680+M*22+L,L*8+M:NEXTM:NEXTL:FORL=0TO30:X=L*2
```

Concurso

```
265 I=33*(RR/33-INT(RR/33))
266 Y=INT(32+31*SIN(.95*I/5)):CH=INT(X/8)*8+INT(Y/8):RO=(Y/8-INT(Y/8))*8:BY=7168
+8*CH+RO
267 BI=7-(X-(INT(X/8)*8)):POKEBY,PEEK(BY)OR(2↑BI):RR=RR+1:NEXTL
268 FORL=0TO7000:NEXT:F=RR-31
269 FORL=0TO30:X=L*2:S=23*(F/23-INT(F/23))
270 Y=INT(32+31*SIN(1.4*S/5)):CH=INT(X/8)*8+INT(Y/8):RO=(Y/8-INT(Y/8))*8:BY=7168
+8*CH+RO
280 BI=7-(X-(INT(X/8)*8)):POKEBY,PEEK(BY)OR(2↑BI):F=F+1:NEXTL:SS=F-31
281 FORT=0TO7000:NEXT
284 FORM=0TO30:X=M*2:E=28*(SS/28-INT(SS/28))
285 Y=INT(32+31*SIN(1.14*E/5)):CH=INT(X/8)*8+INT(Y/8):RO=(Y/8-INT(Y/8))*8:B=7168
+8*CH+RO
286 BI=7-(X-(INT(X/8)*8)):POKEB,PEEK(B)OR(2↑BI):SS=SS+1:NEXTM
287 FORM=0TO30:X=M*2+1:FORY=1TO62
288 CH=INT(X/8)*8+INT(Y/8):RO=(Y/8-INT(Y/8))*8:B=7168+8*CH+RO
289 BI=7-(X-(INT(X/8)*8)):POKEB,PEEK(B)OR(2↑BI):NEXTY:NEXTM
290 FORX=0TO63:Y=32
291 CH=INT(X/8)*8+INT(Y/8):RO=(Y/8-INT(Y/8))*8:B=7168+8*CH+RO
292 BI=7-(X-(INT(X/8)*8)):POKEB,PEEK(B)OR(2↑BI):NEXTX:FORL=0TO27000:NEXT:POKE368
69,240
300 PRINT"]QOTRO MES S=SI N=NO"
339 GETA$
340 IFA$="S"THEN250
341 IFA$<>"N"THEN339
342 GOTO1000
350 IFM$="ENERO"ORM$="1"THENM=1
360 IFM$="FEBRERO"ORM$="2"THENM=2
370 IFM$="MARZO"ORM$="3"THENM=3
380 IFM$="ABRIL"ORM$="4"THENM=4
390 IFM$="MAYO"ORM$="5"THENM=5
400 IFM$="JUNIO"ORM$="6"THENM=6
410 IFM$="JULIO"ORM$="7"THENM=7
420 IFM$="AGOSTO"ORM$="8"THENM=8
430 IFM$="SEPTIEMBRE"ORM$="9"THENM=9
440 IFM$="SETIEMBRE"THENM=9
450 IFM$="OCTUBRE"ORM$="10"THENM=10
460 IFM$="NOVIEMBRE"ORM$="11"THENM=11
470 IFM$="DICIEMBRE"ORM$="12"THENM=12:RETURN
475 Y=Y/4:Z=INT(Y):IFY-Z=0THEN479
476 RETURN
479 X(2)=X(2)+1:RETURN
480 DATA31,28,31,30,31,30,31,31,30,31,30,31
490 PRINT"]":FORO=1TO23:PRINT"||||||||||||||||||||||":NEXTO
495 FORG=0TO20:POKE7922+G,45:NEXT:POKE7701,1:POKE7943,3:POKE8185,2
500 POKE36879,8:FORN=0TO20:I=33*(F/33-INT(F/33)):S=23*(F/23-INT(F/23))
510 E=28*(F/28-INT(F/28))
520 POKE7922+N+22*INT(11*SIN(I*.63*π/10.5)),9
530 POKE7922+N+22*INT(11*SIN(S*.9*π/10.5)),6
540 POKE7922+N+22*INT(11*SIN(E*.75*π/10.5)),5
550 F=F+1:NEXTN
560 FORQ=0TO25000:NEXT
570 PRINT"]QQOTROS 21 DIAS SI=S    NO=N"
580 GETA$:IFA$="S"THEN490
590 IFA$<>"N"THEN580
600 PRINT"]QDE NUEVO S=SI N=NO"
610 GETA$:IFA$="S"THENF=F-21:GOTO490
620 IFA$<>"N"THEN610
1000 POKE789,0
```

SU PROGRAMA PARA CUALQUIER SISTEMA COMMODORE PUEDE HACERLE GANAR 5.000 PTAS.

EL PRESENTE CONCURSO ESTA ABIERTO A TODOS NUESTROS LECTORES Y SU PARTICIPACION E INSCRIPCION ES GRATUITA.
LEA LAS BASES DEL CONCURSO

- NO SE ESTABLECEN LIMITACIONES EN CUANTO A EXTENSION, TEMA ELEGIDO O MODELO DE ORDENADOR
- LOS CONCURSANTES DEBERAN ENVIARNOS A LA DIRECCION QUE FIGURA AL PIE, EL CASSETTE O DISKETTE CONTENIENDO EL PROGRAMA, UNA EXPLICACION DEL MISMO Y, AL SER POSIBLE, UN LISTADO EN PAPEL DE IMPRESORA. SE PODRAN ENVIAR TANTOS PROGRAMAS COMO SE DESEE
- LOS PROGRAMAS, PREVIA SELECCION, SERAN PUBLICADOS EN LA REVISTA, OBTENIENDO TODOS ELLOS 5.000 PTAS.
- LA DECISION SOBRE LA PUBLICACION O NO DE UN PROGRAMA CORRESPONDE UNICAMENTE AL JURADO NOMBRADO AL EFECTO POR "COMMODORE MAGAZINE", SIENDO SU FALLO INAPELABLE
- LOS CRITERIOS DE SELECCION SE BASARAN EN LA CREATIVIDAD DEL TEMA ELEGIDO Y LA ORIGINALIDAD Y/O SENCILLEZ EN EL METODO DE PROGRAMACION GLOBAL
- ENVIAR A:
CONCURSO COMMODORE MAGAZINE



commodore Magazine

C/Bravo Murillo, 377. 28002-Madrid -

Concurso

Othello 64

PREMIADO CON 5.000 PESETAS

Esta interesante versión del OTHELLO para el Commodore 64 nos llega nada menos que de la isla de Tenerife y nos la envía Victor Castro Reyes. Para los que no lo conozcan diremos que el Othello es un juego de estrategia que se desarrolla sobre un tablero de ajedrez de 8 × 8 casillas blancas y negras. El juego se desarrolla entre dos jugadores que disponen de 64 fichas, blancas por una cara y negras por la otra, que deben ir colocando en el tablero por turno. Para colocar una ficha de su color, el jugador al que le corresponda el turno, debe buscar una casilla de tal forma que al colocar su ficha queden encerradas fichas del contrario (del otro color) entre dos fichas del jugador, tanto en horizontal o vertical como en diagonal. Si un jugador no puede colocar ninguna ficha pasa su turno y juega el contrario. El juego termina cuando ninguno de los jugadores puede colocar más fichas o cuando se completa el tablero, declarándose ganador el jugador que tenga más fichas de su color. Todas las fichas del contrario que quedan encerradas entre dos fichas al realizar cada jugada se dan la vuelta y pasan a ser del jugador que las encerró.

Las características del programa OTHELLO 64 son las siguientes:

Juega a un nivel bastante bueno teniendo en cuenta al jugar una ficha tanto su posición como el número de fichas contrarias capturadas.

Para realizar la jugada se emplea un sistema de coordinadas cartesianas XY, donde tanto X como Y pueden ser un número o una letra.

El programa rechaza las jugadas ilegales.

Si se desea abandonar la partida antes de terminar se puede hacer pulsando la tecla ''asterisco'' seguida de RETURN en lugar de la jugada.

La partida termina automáticamente indicando el resultado tanto cuando se completa el tablero como cuando se da la circunstancia de que ninguno de los jugadores pueda colocar nuevas fichas.

Para los interesados, la estructura del programa es la siguiente:

2Ø — 11Ø	Analisis de jugada válida y piezas capturadas
12Ø — 23Ø	Dibujo del tablero
24Ø — 38Ø	Juega y presenta su jugada el ordenador
39Ø — 56Ø	Jugada del jugador, chequeo de validez y presentación
56Ø — 64Ø	Transferencia de caracteres y definición de caracteres
65Ø — 66Ø	Dimensiona matriz, y asigna valor estratégico a cada casilla
68Ø — 74Ø	Sentencias DATA
79Ø — 87Ø	Presentación de resultados y vencedor

```
10 CLR:PRINT "▮▮▮▮","UN MOMENTO, POR FAVOR":GOTO 570
20 REM ***ANALISIS***
30 A=0:B=A:IF ABS(A(U))=H THEN RETURN
40 FOR I=H TO 8:L=H:K=U
50 K=K+N(I):IF ABS(A(K))<>H THEN 110
60 IF A(K)=F THEN 80
70 L=L+H:GOTO 50
80 A=A+L-H:IF B<A THEN B=A
90 IF R=0 THEN 110
100 FOR L=0 TO A:A(U+L*N(I))=F:NEXT L:A=0
110 NEXT I:RETURN
120 REM ***TABLERO***
130 S=0:C=S
140 PRINT"▮":PRINT,"A B C D E F G H"
150 PRINT ,"▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔┐"
160 FOR X=H TO 8:PRINT:PRINT TAB(7)X;
170 FOR Y=H TO 8:Z=A(X+10*Y)
180 IF Z=-H THEN PRINT "▔▁▮▮▮▮ ▮┐";:S=S+H
190 IF Z=H THEN PRINT "♠|▮▮▮─┐";:C=C+H
200 IF ABS(Z)<>H THEN PRINT"┌▮▮▮▮▁┐";
210 NEXT Y:PRINT X:NEXT X
220 POKE 781,10:POKE 782,30:POKE 783,0:SYS 65520:PRINT "TIENES   ▮▮▮▮";C;"▮▮"
230 PRINT:PRINT,,"TENGO    ▮▮▮▮";S:RETURN
240 REM ***JUEGA CMB-64***
250 M=0:R=M:F=-H
```

SERVICIO DE EJEMPLARES ATRASADOS



Complete su colección de COMMODORE MAGAZINE.

A continuación le resumimos el contenido de los ejemplares aparecidos hasta ahora.

Núm. 1 - 250 Ptas.
Análisis de la nueva serie 700/Calc result. a fin de cuentas/Más potencia con Victree/Cómo adaptar cualquier cassette/Juegos y aplicaciones para VIC-20 y CBM 64.

Núm. 2 - 250 Ptas.
CBM 64 en profundidad/Superbase 64: el ordenador que archiva/Juegos. trucos y aplicaciones.

Núm. 3 - 250 Ptas.
Magic Desk, el despacho en casa/Herramientas para el programado/Interfaces para todos.

Núm. 4 - 250 Ptas.
El 64 transportable revisado a fondo/Interface RS 232 para el VIC-20/Juegos/El fútbol-silla en su salón.

Núm. 5 - 250 Ptas.
Programas, juegos y concurso/Londres: Quinta feria Commodore/Basic, versión 4.75.

Núm. 6 - 250 Ptas.
El misterio del Basic/Lápices ópticos para todos/Concurso. juegos, aplicaciones.

Núm. 7 - 250 Ptas.
El ordenador virtuoso. MusiCalc. Programa monitor para el 64. Lápices ópticos. Ampliación de memoria para Vic-20.

Núm. 8 - 250 Ptas.
Joystick y Paddle para todos. Misterio del BASIC. EL LOGO. Cálculo financiero. Programas.

Núm. 9 - 250 Ptas.
Conversión de programas del Vic-20 al C-64. Móntate un paddle. Identifica tus errores. Software comentado.

Núm. 10 - 250 Ptas.
Koala Pad: La potencia de un paquete gráfico. Trucos. El FORTH. Software comentado. El LOGO.

Corte y envíe este cupón a: COMMODORE MAGAZINE
Bravo Murillo, 377 - Tel. 733 96 62 - 28020-MADRID

SERVICIO DE EJEMPLARES ATRASADOS

Ruego me envíen los siguientes ejemplares atrasados de COMMODORE MAGAZINE:

____

El importe lo abonaré:
Contra reembolso □ Adjunto Cheque □ Con mi tarjeta de crédito □
American Express □ Visa □ Interbank □ Fecha de caducidad: ________

Número de mi tarjeta: □□□□□□□□□□□□□□□□□□□□□□□□□□

NOMBRE ______________________________

DIRECCION ______________________________

CIUDAD ______________________________ D.P. ________

PROVINCIA ______________________________

Concurso

```
260 FOR X=H TO 8:FOR Y=H TO 8
270 U=X+10*Y:GOSUB 30
280 IF B>0 AND B+A(U)=M AND RND(1)>.5 THEN Z=U
290 IF B>0 AND B+A(U)>M THEN M=B+A(U):Z=U
300 NEXT Y:NEXT X
310 IF M=0 THEN 330
320 R=H:U=Z:GOSUB 30
330 POKE 781,22:POKE 782,15:POKE 783,0:SYS65520
340 IF M=0 THEN PRINT "PASO!!!":GOTO 360
350 PRINT "YO MUEVO  ";Z-10*INT(Z/10);"▌";CHR$(Z/10+64)
360 GOSUB 130
370 IF H$="" AND M=0 THEN 790
380 IF S+C=64 OR C=0 THEN 790
390 REM ***JUEGAS***
400 POKE 781,21:POKE 782,15:POKE 783,0:SYS65520:PRINT "TU MUEVES     ████▌";
410 H$="":INPUT H$: H$=LEFT$(H$,2)
420 POKE 781,21:POKE 782,24:POKE783,0:SYS65520: PRINT "  ";H$;
430 IF H$=""THEN 540
440 IF ASC(H$)>64 THEN H$=H$+LEFT$(H$,H):H$=RIGHT$(H$,2)
450 IF H$="*" THEN 790
460 X=ASC(H$)-48:IF X<H OR X>8 THEN 400
470 Y=ASC(RIGHT$(H$,H))-64:IF Y<H OR Y>8THEN 400
480 U=X+10*Y:F=H:R=0: GOSUB 30
490 IF A=0 THEN 400
500 R=H:GOSUB 30:GOSUB 130
510 IF S+C=64 OR S=0 THEN 790
520 POKE 781,22:POKE 782,15:POKE 783,0:SYS65520:PRINT "MUEVO YO: ▒RETURN▀";
530 INPUT H$
540 POKE 781,22:POKE 782,15:POKE 783,0:SYS65520:PRINT "                    "
550 GOTO 250
560 REM ***UDG***
570 POKE 56334,PEEK(56334)AND254
580 POKE 1,PEEK(1)AND251
590 FOR I=0 TO 1535
600 POKE I+12288,PEEK(I+53248)
610 NEXT I:POKE 1,PEEK(1)OR4
620 POKE 56334,PEEK(56334)OR1
630 POKE 53272,(PEEK(53272)AND240)+12
640 FOR N=12808 TO 12871:READ H:POKE N,H:NEXTN
650 DIM A(99):FOR I=1 TO 8:READ N(I):NEXT I
660 FOR I=11 TO 88:READ A(I):NEXT I
670 REM ***DATAS***
680 DATA 0,3,15,31,63,63,127,127,0,192,240,248,252,252,254,254,127,127,63,63,31
690 DATA 15,3,0,254,254,252,252,248,240,192,0,255,252,240,224,192,192,128,128
700 DATA 255,63,15,7,3,3,1,1,128,128,192,192,224,240,252,255,1,1,3,3,7,15,63,255
710 DATA 1,-1,9,-9,10,-10,11,-11
720 DATA 9,.9,8,6,6,8,.9,9,0,0,.9,.1,2,2,2,2,.1,.9,0,0,8,2,8,6,6,8,2,8,0,0
730 DATA 6,2,6,1,-1,6,2,6,0,0,6,2,6,-1,1,6,2,6,0,0,8,2,8,6,6,8,2,8,0,0
740 DATA .9,.1,2,2,2,2,.1,.9,0,0,9,.9,8,6,6,8,.9,9
750 PRINT"▯":H=1:GOSUB 130
760 POKE 781,9:POKE 782,1:POKE 783,0:SYS65520:PRINT "♠|█▯▯─¯ TU"
770 PRINT:PRINT" ¯▁█▯▯| | YO":GOTO 400
780 REM ***FINAL***
790 POKE 781,21:POKE 782,15:POKE 783,0:SYS65520
800 PRINT"
810 POKE 781,21:POKE 782,9:POKE 783,0:SYS65520
820 IF S>C THEN PRINT "GANO YO"
830 IF S<C THEN PRINT "ENHORABUENA, GANAS TU"
840 IF S=C THEN PRINT "EMPATE A";S
850 PRINT TAB(9)"OTRA PARTIDA S/N?"
860 GET A$:IF A$<>"S" AND A$<>"N" THEN 860
870 IF A$="S" THEN PRINT"▯":RUN 640
```

PROGRAMA commodore Magazine COMPROBADO

GUIA PRACTICA



HACEMOS FACIL LA INFORMATICA

- SINCLAIR • SPECTRAVIDEO
- COMMODORE • DRAGON
- AMSTRAD • APPLE
- SPERRY UNIVAC

Modesto Lafuente, 63 Telf. 253 94 54 28003 MADRID	Colombia, 39-41 Telf. 458 61 71 28016 MADRID
José Ortega y Gasset, 21 Telf. 411 28 50 28006 MADRID	Padre Damián, 18 Telf. 259 86 13 28036 MADRID
Fuencarral, 100 Telf. 221 23 62 28004 MADRID	Avda. Gaudí, 15 Telf. 256 19 14 08015 BARCELONA
Ezequiel González, 28 Telf. 43 68 65 40002 SEGOVIA	Stuart, 7 Telf. 891 70 36 ARANJUEZ (Madrid)

PARA COMMODORE 64

SI DESEA TENER COPIA DE SEGURIDAD DE SUS JUEGOS O PROGRAMAS EN CINTA LO PODRA REALIZAR CON EL PROGRAMA "COPY-SEGUR". PRECIO DE LA CINTA **2.000** PESETAS.
ELIJA FORMA DE PAGO: TALON BANCARIO LIBRE DE GASTOS. CONTRARREEMBOLSO MAS 200 PESETAS POR GASTOS DE ENVIO.

PEDIDOS AL APARTADO 9.032 DE VALENCIA

ELECTRONICA SANDOVAL S.A.

* * * * * * * * * * * * * *

DISTRIBUIDORES DE:

COMMODORE-64
ORIC-ATMOS
ZX SPECTRUM
SINCLAIR ZX 81
ROCKWELL'-AIM-65
DRAGON-32
NEW BRAIN
DRAGON-64
CASIO FP-200

* * * * * * * * * * * * * *

ELECTRONICA SANDOVAL, S. A.
C/. SANDOVAL 3, 4, 6 - MADRID-10
Teléfonos: 445 75 58-445 76 00-445 18 70-447 42 01

COMPARE LOS PRECIOS

COMMODORE 64	69.000 pts.
DATASSETTE	12.000 pts.
FAST TURBO MENUE	8.500 pts.
SOLO FLIBHT	3.900 pts.
COLOSSUS CHESS 2.0	2.800 pts.
ZAXXON	2.300 pts.
	98.500 pts.

POR SOLO **69.900 pts.**

MODEM COMPLETO DESDE	16.000 pts.
COMMODORE SX-64	180.000 pts.
FLOPPY	63.500 pts.
COMMODORE 64	57.500 pts.
IMPRESORA MP 2080	63.500 pts.

PLAZAS LIBRES PARA SU DISTRIBUCION EN PROVINCIAS

ASTOC-DATA
Hardware y Software-Systems
Sarela de Abajo Santiago de Compostela Tel. 59 95 33

ULTIMO AVISO

¿Eres aficionado a la programación?
¿Dominas el código máquina?
¿Tienes programas originales?
¿Puedes escribir un buen juego?
¿Quieres ganar dólares, libras, francos o pesetas desde tu casa, en tus horas libres?

¡NO TE LO PIERDAS!

Contacta inmeditamente con:

CIBERCOMP, S. A.
Tels. (91) 200 21 00
(91) 759 22 44

Especialistas en software para Home Computers, asociados con primeras firmas internacionales.



COMMODORE 64
SPECTRUM
CASIO

PRECIOS INTERESANTES

CMP

Arturo Soria, 154
Tel. 415 93 28
28043 MADRID

TRONIK

Bigay, 11-13
Tel. (93) 212 85 96
Barcelona-22

¡HOLA, SOY **TRONIK** TU AMIGO INFORMATICO!

- Todo sobre el **COMMODORE 64** y **VIC 20**
- Periféricos.
- Múltiples programas.
- Libros y revistas.
- Recompramos tu ordenador como entrada de otro nuevo.
- Cursillos de BASIC a todos los niveles.

CAMAFEO INC.

CASSETTES DE CALIDAD PROBADA PARA ORDENADORES

Cada uno	Caja de 10	Caja de 30
C-5 199 ptas.	1.393 ptas.	3.582 ptas.
C-10 209 ptas.	1.463 ptas.	3.762 ptas.
C-15 219 ptas.	1.533 ptas.	3.942 ptas.
C-20 229 ptas.	1.602 ptas.	4.122 ptas.

Libre de gastos de envío contra reembolso correos

CAMAFEO INC.
José Lázaro Galdiano, 1. 28036 Madrid.

DEFOREST
MICROINFORMATICA

TODO SOBRE **COMMODORE** - **64** Y **VIC** - **20**

LOS ULTIMOS JUEGOS EN EL MERCADO
TODO EN PERIFERICOS - LIBROS
PROGRAMAS DE GESTION - ETC.

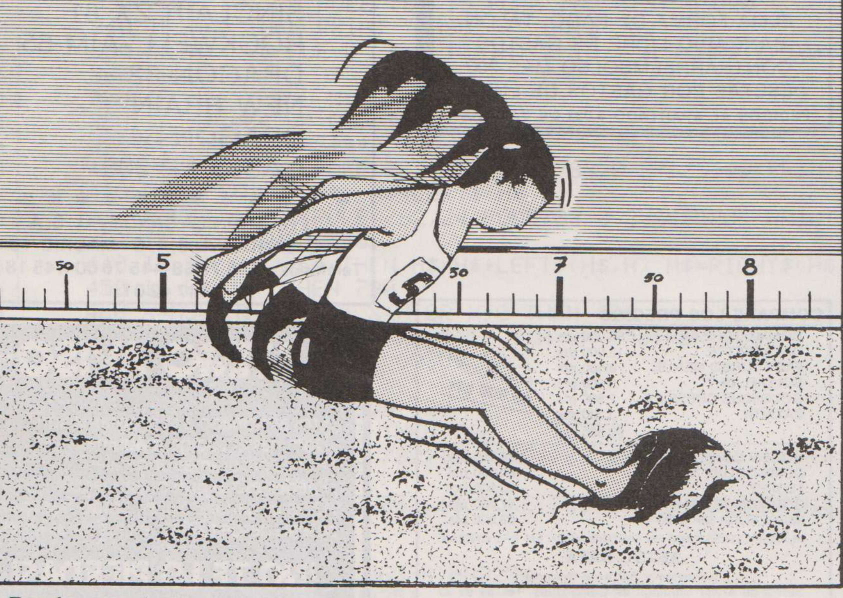
SOLICITE INFORMACION POR CORREO

BARCELONA-15
C/ Viladomat, 105. Tel. 223 72 29

Concurso

Salto de longitud

VIC-20

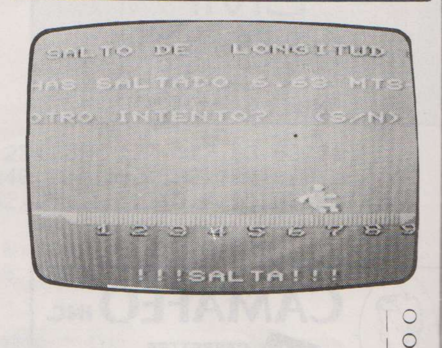


Desde que el pasado verano se celebraron los juegos olímpicos en la ciudad de Los Angeles, han proliferado los juegos de ordenador que representan pruebas de atletismo. SALTO DE LONGITUD, programa que nos envía Gabriel Torres desde la isla de Mallorca, es una versión de este tipo de programas diseñada para un VIC-20 en su versión estandar.

El objetivo del jugador es batir el record del salto de longitud. Para ello debe conseguir la mayor velocidad posible en la carrera que precede al salto. Inicialmente, el saltador espera al pie de la pista a que los jueces den la salida. A partir de ese momento debe empezar a correr tan deprisa como pueda hasta llegar al punto de salto. Para hacer correr al atleta, el jugador debe pulsar repetidamente la tecla F7. Cuanto más deprisa pulse la tecla, mayor será la velocidad del atleta y más largo el salto. Las marcas obtenidas se encontrarán alrededor de los 8 metros, siendo el record de la redacción de **Commodore Magazine** de unos 8,10 metros, ¡a ver si lo superáis!

La estructura del programa es como sigue:

1-60	Presentación e instrucciones de juego.
70	Envía subrutina de formación de caracteres.
90-200	Presentación de la pista y del atleta.
220-360	Carrera del atleta.
1000-1010	Subrutina de la señal de comienzo.
1400-1410	Subrutina de ruido de pasos para la carrera.
1500-1660	Subrutina de salto.
2000-2020	Subrutina de borrado del atleta.
3000-3060	Presentación del resultado obtenido y final.
9000-10005	Subrutina de formación de caracteres del atleta.




```
1 REM SALTO LONGITUD
5 PRINT"■":PRINT"■          ATLETISMO       "
10 PRINT"■■■■■■■■■■■■■■■ ◠◣◠■◠"
12 PRINT"■■■■■■■■■ ◡◣◡■◡"
14 PRINT"■■■■■■■■■■◠■◠"
16 PRINT"■■■■■■■■■■◡■◡"
18 PRINT"■■■■■■■■■■BARCELONA*92"
20 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■◣G.T.R."
22 FORI=1TO5000:NEXT
25 PRINT"■":POKE36879,109
30 PRINT"■■* SALTO DE  LONGITUD *■"
40 PRINT"■■■■■■■■■■■■  HAZ CORRER CON F.7";
50 PRINT"■■  AL SALTADOR LO MAS"
```

```
60 PRINT"QQ  RAPIDO QUE PUEDAS.■"
70 GOSUB9000
90 POKE36869,255:POKE36879,110
100 PRINT"■"
110 PRINT"■■* SALTO DE  LONGITUD *■"
120 A$="▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔"
125 C$="■←■■■■■■■■■■■■ - "
130 D$="■← ■■■■!■£■■■■■■■■%&"
135 E$="■← ■■■■■■@■↑■■■■ []"
140 F$="                    "
160 PRINT"QQQQQQQQQQQQQQ"TAB(3)C$
170 PRINT"■■"A$;
180 PRINT"QQQQQ■■■ESPERA LA  SENAL"
190 PRINT"■]"TAB(16)CHR$(18)CHR$(102)
200 FORI=1TO6000:NEXT:GOSUB1000
220 A$=A$+"▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔▔"
240 A$=A$+"▔▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▔"
250 PRINT"QQQQQQQQQQQQQQQQQQQQQQ        ■■A CORRER!!!     ":POKE8136,32
260 TI$="000000":W=0
270 GETB$:IFB$=""THEN270
280 IFB$<>"■"THEN270
290 W=W+1
300 PRINT"QQQQQQQQQQQQQQ"TAB(3)D$
310 GOSUB1400:FORI=1TO50:NEXT
320 PRINT"■■"MID$(A$,W,22)
330 PRINT"QQQQQQQQQQQQQQ"TAB(3)C$
340 GOSUB1400:FORI=1TO50:NEXT
350 IFW=43THEN1500
360 GOTO270
1000 POKE36878,15:POKE36876,231:FORT=1TO100:NEXTT:POKE36878,0:POKE36876,0
1010 RETURN
1400 POKE36878,5:POKE36877,130:FORM=1TO10:NEXT
1410 POKE36878,0:RETURN
1500 L=10-(TI/325):PRINT"QQQQQQQQQQQQQQQQQQQQQ■■■■■1 2 3 4 5 6 7 8 9";
1510 PRINT"QQQQQQQQQQQQQQQQQQQQQQ■        !!!SALTA!!!■"
1550 GOSUB2000:PRINT"QQQQQQQQQQQQ"TAB(4)D$
1560 FORI=1TO100:NEXT
1570 GOSUB2000:PRINT"QQQQQQQQQQQQ"TAB(5)D$
1570 GOSUB2000:PRINT"QQQQQQQQQQQQ"TAB(5)D$
1580 FORI=1TO100:NEXT
1600 FORJ=6TOINT(L*2):GOSUB2000:PRINT"QQQQQQQQQQQQ"TAB(J)D$
1610 FORI=1TO100:NEXT:GOSUB2000:NEXTJ
1620 PRINT"QQQQQQQQQQQQQ"TAB(J+1)E$
1630 FORI=1TO100:NEXT:GOSUB2000
1640 PRINT"QQQQQQQQQQQQQQ"TAB(J+2)E$
1660 GOTO3000
2000 PRINT"QQQQQQQQQQQQ"F$
2010 PRINTF$:PRINTF$:PRINTF$:PRINTF$
2020 RETURN
3000 PRINT"QQQQQ■-HAS SALTADO"L
3010 PRINT"QQQQQ"TAB(17)"■ MTS-"
3015 FORI=1TO6000:NEXT
3020 PRINT"QQQQQQQQ■ OTRO INTENTO?  (S/N)"
3030 GETK$:IFK$=""THEN3030
3040 IFK$="S"THEN100
3045 IFK$<>"N"THEN3030
3050 POKE36869,240:POKE36879,27:PRINT"■QQQQQQQQQQ■   !HASTA OTRA,AMIGO!
3060 PRINT:PRINT:PRINT:END
9000 DATA 31 , 0, 0, 0, 28, 30, 28, 28, 0
```

PREMIADO CON 5.000 PESETAS

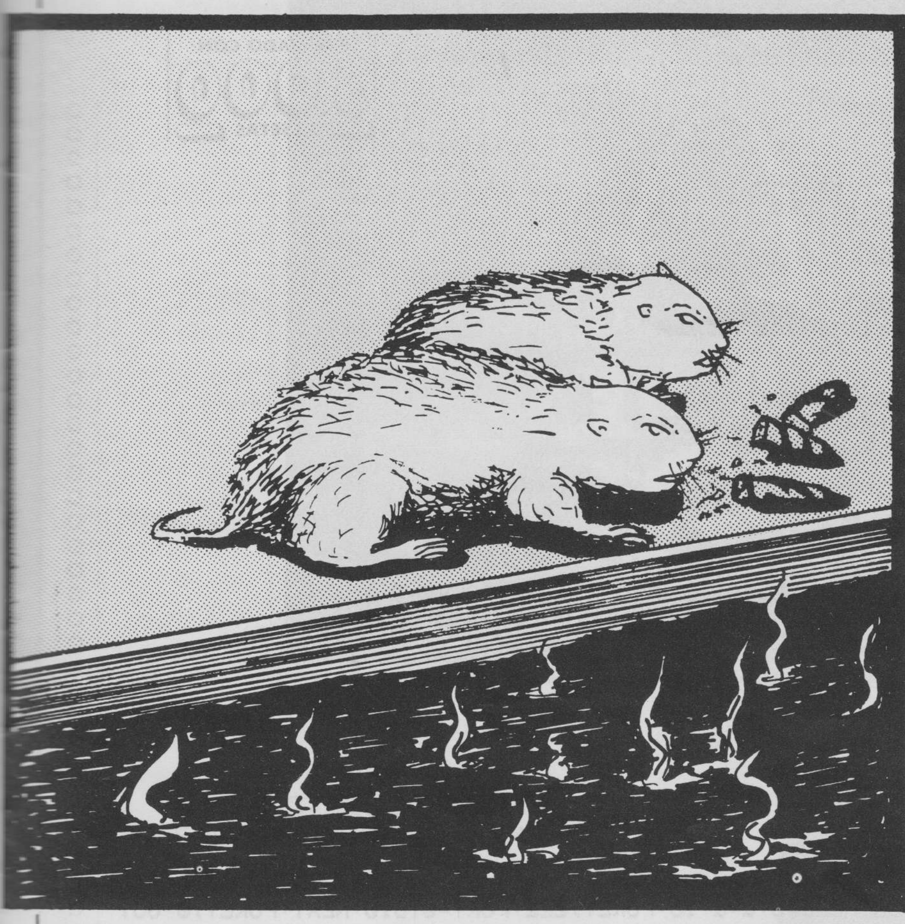
Concurso

```
9001 DATA 33 , 7, 7, 6, 6, 14, 6, 0, 0
9002 DATA 28 , 252, 254, 255, 127, 126, 126, 126, 126
9003 DATA 35 , 0, 16, 248, 248, 0, 0, 0, 0
9004 DATA 36 , 1, 3, 7, 14, 28, 56, 32, 32
9005 DATA 37 , 255, 255, 240, 0, 0, 0, 0, 0
9006 DATA 38 , 240, 240, 48, 48, 56, 56, 0, 0
9007 DATA 39 , 0, 0, 0, 48, 112, 48, 48, 48
9008 DATA 40 , 0, 0, 1, 7, 7, 14, 6, 0
9009 DATA 41 , 62, 127, 255, 254, 126, 126, 126, 126
9010 DATA 43 , 48, 240, 240, 0, 0, 0, 0, 0
9011 DATA 45 , 126, 127, 99, 103, 238, 204, 238, 238
9012 DATA 0 , 62, 126,255, 255, 126, 126, 126, 127
9013 DATA 30 ,0, 32, 240, 240, 0, 0, 0, 0
9014 DATA 27 , 63, 63, 28, 14, 7, 7, 3, 0
9015 DATA 29 , 240, 240, 48, 48, 56, 56, 0, 0
10000 G1=32768:G2=7168:FORI=0TO511
10001 POKEG2+I,PEEK(G1+I)
10002 NEXTI
10003 FORI=1TO 16: READC:FORJ=0TO7:READK
10004 POKEG2+(C*8)+J,K:NEXTJ,I:POKE52,28:POKE56,28
10005 RETURN
```



Ratas

VIC-20



Jaume María Saguer nos manda, desde Gerona, este terrorífico programa llamado RATAS, diseñado para funcionar en un VIC-20 sin expansiones de memoria de ningún tipo. En una desolada mansión de Transilvania está ocurriendo algo terrible. Se trata de una invasión de ratas, que llegan a todas partes en grandes cantidades y corretean por las paredes hacia el suelo por el que desaparecen, escabulléndose por rendijas y recovecos del pavimento. Los habitantes de la mansión, decididos a acabar con tan terrible invasión, han adquirido un estupendo gato (el jugador) que debe comerse todas las ratas que pueda.

Las ratas, además de aterrorizar a todo el vecindario, están dispuestas a comerse el queso destinado al gato.

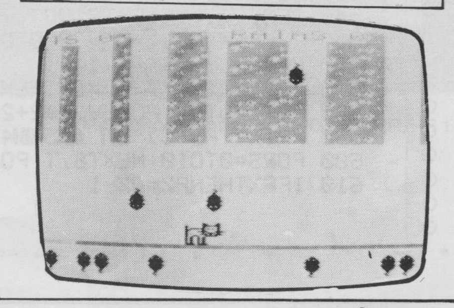
El gato-jugador debe comerse a las ratas de la siguiente forma. Mediante las teclas «:» e «=» debe desplazarse a derecha e izquierda hasta situarse debajo de alguna rata. Cuando esta esté lo suficientemente cerca, el gato deberá dar un salto (pulsando la barra de espacios) con lo que se zampará la rata.

Cada rata eliminada aumentará la puntuación del gato. Otro elemento que permitirá aumentar la puntuación es un plato de leche, que aparecerá de forma esporádica en algún lugar del suelo, y que el gato deberá intentar alcanzar.

Las ratas, al llegar al suelo, buscarán un agujero por el que colarse hacia el sótano. Por cada rata que alcance su objetivo el gato perderá unos cuantos puntos. Además, las ratas en el sótano se vuelven peligrosas, y pueden zamparse al gato si este no se anda con cuidado. Esto ocurre cuando el gato se sitúa descuidadamente encima del agujero del suelo. El gato por otra parte es como todos los gatos, tiene varias vidas, (en este caso tres).

Pasando a cuestiones más serias, os adjuntamos la estructura del programa, a grandes rasgos, y una lista de algunas de las variables utilizadas por el mismo.

L%	Vidas.
F	Dirección del agujero.
W	Dirección del gato.
V	Velocidad del gato.
C$	Nombre del queso en pantalla.
PA	Número de pantalla (nivel).
P	Posición del gato.
C	Posición del agujero.
130-170	Movimiento de las ratas.
180-310	Movimiento del gato.
320-360	Situación del agujero.
370-450	Situación del plato de leche.
460-530	Muerte del gato.
540-630	Entrada de ratas en el agujero.
640-780	Inicialización.
790-910	Presentación y nivel de juego.
1000-1120	Datas de tipos de queso y caracteres (gato y ratas).



Concurso

PREMIADO CON **5.000** PESETAS

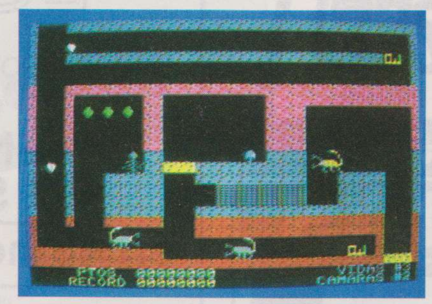
```
100 GOTO640
110 PRINTTAB(P)"■D? ■■■K= ":RETURN
120 PRINTTAB(P)"■ ";■■■■ 89":RETURN
130 REM.........RATS
140 PRINT"■■PTS"Q,"RATAS"A%
150 T=FNH(O%)-1:R%(T)=R%(T)+1
160 IFR%(T)>14THENR%(T)=1:D%(T)=FNH(20)
170 PRINTLEFT$(A$,R%(T))SPC(D%(T))"■ ■■5■■4"
180 REM............GAT
190 T=PEEK(203):PRINTA$
200 IFT=45THENW=1:IFP>0THENP=P-V:POKEY,230
210 IFT=46THENW=2:IFP<18THENP=P+V:POKEY,230
220 P=ABS(P):POKEY,0
230 IFT-32GOTO310
240 PRINTTAB(P)"■■  .TI]"
250 ONWGOSUB110,120
260 IFPEEK(8010+P+W)=32GOTO290
270 PRINTTAB(P+W)".TII] ■■ ■■"
280 Q=Q+5:A%=A%+1:FORT=235TO170STEP-10:POKEY-1,T:NEXT:POKEY-1,0
290 IFA%=10GOTO800
300 FORT=0TO30:NEXT:PRINTTAB(P)"■[I]  ":POKEY,255:POKEY,0
310 ONWGOSUB110,120
320 REM........AGUJERO
330 IF.95<RND(0)THENF=-F
340 IFC<1THENF=1
350 IFC>18THENF=-1
360 C=C+F:PRINTTAB(C)"■■  ─"
370 REM.........PLAT
380 IFZGOTO410
390 IFV+.09>RND(0)GOTO440
400 PO=FNH(18):Z=FNH(20)
410 Z=Z-1:IFPEEK(8076+PO)>55THENZ=0:Q=Q+15:POKEY,222:FORT=0TO10:NEXT:POKEY,0:GOT
0440
420 PRINTTAB(PO)".TI■3■■"
430 IFZ=0THENPRINTTAB(PO)".TII ■■"
440 IFPEEK(8055+C)=32GOTO550
450 IFPEEK(8121+C)-54GOTO140
460 REM.........MUERTE
470 L%=L%-1:PRINTTAB(C)".T■2"
480 FORT=228TO240:POKEY,T:POKEY-1,T+4:NEXT:POKEY-1,0
490 FORT=240TO218STEPNOT0:POKEY,T:NEXT:POKEY,0
500 FORT=0TO500:NEXT
510 IFL%=0GOTO670
520 PRINTA$TAB(P)"■■  ■■■  ":P=0:W=2
530 GOTO140
540 REM.........ENTR.AG
550 IFPEEK(8011+C)=32GOTO140
560 PRINTLEFT$(A$,14)
570 IFPEEK(8077+C)=51THENZ=-1:A%=0:Q=INT(Q/2)
580 FORT=0TO4:POKEY,T*2+222:POKEY-1,222-T
590 PRINTTAB(C)"■■ ■■5■■4.T]"
600 FORS=0TO10:NEXTS,T:POKEY,0:POKEY-1,0
610 IFA%THENA%=A%-1
```

PROGRAMA commodore COMPROBADO

Concurso

```
620 PRINTTAB(C)"█ ██6█7.T"
630 GOTO140
640 REM...........INIT
650 POKE55,0:POKE56,29:FORT=7424TO7432:POKET,0:NEXT:FORT=0TO20:READA$:NEXT
660 FORT=7528TO7679:READA:POKET,A:NEXT
670 CLR:L%=3:F=1:W=2:V=.9:Y=36876:DIMR%(7),D%(7)
680 A$="█████████████████"
690 DEFFNH(T)=INT(T*RND(0)+1)
700 POKEY+3,28:POKEY-7,255:POKEY+2,15
710 PRINTCHR$(8)"█"RIGHT$(A$,4)TAB(7)"█:;  >?":PRINTTAB(7)"89  <="
720 PRINTTAB(6)"███--------"
730 PRINTTAB(6)" ███RATS"
740 PRINTTAB(6)"█---------"
750 PRINTRIGHT$(A$,7)"███APRIETA UNA TECLA."
760 PRINT"███PARA TERMINAR [E]"
770 POKE198,0:WAIT198,1:GETC$
780 IFC$="E"THENPOKE0,0:SYS0
790 REM...........PANT
800 PRINT"███NIVEL"PA
810 PA=PA+1:A%=0:READC$,D1,D2
820 PRINTRIGHT$(A$,9)SPC(5)"██"C$:IFC$="█CHESTER"THENRESTORE
830 O%=O%+1:IFO%>7THENO%=1:V=V-.3
840 IFV<.2THENV=.2
850 FORT=0TO1000:NEXT
860 FORT=46TO0STEPNOT1:POKEY-9,T:NEXT:PRINT"█";
870 FORT=1TO220/D2:PRINTMID$("0.1/",D1,D2);:NEXT
880 PRINTA$"████──────────────"
890 POKEY-9,46
900 FORT=0TOO%-1:R%(T)=0:D%(T)=FNH(20):NEXT
910 GOTO140
1000 DATA"█CAMEMBERT",1,4,GRUYERE,3,1,EMMENTAL,2,2,MANCHEGO,3,2,ROQUEFORT,4,1
1010 DATA"█EDAM",1,2,"█CHESTER",2,1
1020 REM.........CHARS
1030 DATA 0,0,0,85,170,0,0,0
1040 DATA255,253,248,205,135,135,207,255,255,255,255,255,251,241,251,255
1050 DATA255,255,255,159,79,47,159,255,251,255,191,31,191,255,251,245
1060 DATA16,56,84,16,56,16,56,16,0,0,0,0,255,255,126,60
1070 DATA 254,124,124,254,84,124,56,16,16,8,16,16,56,124,124,124
1080 DATA16,56,84,124,254,124,124,254,124,124,56,16,16,32,16,0
1090 DATA131,152,165,165,165,165,165,247,174,248,0,0,0,0,0,128
1100 DATA1,1,129,129,191,193,135,129,4,140,252,4,84,4,143,116
1110 DATA117,31,0,0,0,0,0,1,193,25,165,165,165,165,165,239
1120 DATA32,49,63,32,42,32,241,46,128,128,129,129,253,131,225,129
```

PROGRAMA ★ commodore Magazine ★ COMPROBADO



PROGRAMA: FIRE ANT
TIPO: JUEGO
DISTRIBUIDOR: SOFTWARE ESPAÑA
ORDENADOR: COMMODORE 64 CON JOYSTICK

Aunque la mayoría de los juegos se basan en cohetes, marcianos y otros adelantos de la técnica, algunos programadores recurren a elementos más cotidianos para obtener la inspiración de sus juegos. Este es el caso del programa FIRE ANT, en el que una heroica hormiga debe introducirse en un nido de escorpiones para rescatar a la hormiga reina, que ha sido secuestrada por éstos.

La carátula de la cinta, conteniendo las instrucciones de juego, está impresa en español y, para mayor sorpresa nuestra, el programa también está en nuestro idioma. Una vez leído, se nos pregunta si queremos ver las instrucciones o no y, a continuación, empieza el juego. En la pantalla que le precede, vemos a nuestra hormiga en compañía de la reina, por detrás aparece un escorpión que nos empuja y secuestra a la dama llevándosela a su nido. Después de esta introducción empieza realmente el juego, viéndose en pantalla un corte en sección de la primera parte del nido de los escorpiones. Para poder seguir bajando y llegar a la octava sección, donde se encuentra la reina, debemos realizar en cada nivel una serie de acciones que nos dejen el paso libre. En todos ellos existen unas puertas amarillas que debemos abrir con una llave dispuesta a tal efecto (una distinta para cada puerta, y una vez abierta desaparece). Esta llave tiene la curiosa propiedad de que si la llevamos encima no podemos coger ninguna otra cosa y viceversa, es decir, si llevamos algo no podemos coger la llave; además los objetos que cogemos no los podemos soltar hasta que los usemos y, por tanto, si cogemos la llave antes de tiempo tendremos que volver a empezar desde el principio. Algunas de las acciones que debemos realizar son: en el primer nivel hay que abrir una trampilla (no una puerta) cogiendo una jarra y subiendo hasta la salida; en ese momento cae un chorro de agua que rompe la trampilla situada debajo; después de esto debemos coger una llave que nos abre una puerta para coger otra y poder abrir la puerta de salida (como ya comentamos, cada llave abre una puerta y no las podemos intercambiar). En el segundo nivel el proceso es más complicado.

PUNTUACION:
ADICION: 8
PRESENTACION: 6
GRAFICOS: 6
ACCION: 7

PROGRAMA: ANNIHILATOR
TIPO: JUEGO
DISTRIBUIDOR: SOFTWARE ESPAÑA
ORDENADOR: COMMODORE 64 CON JOYSTICK

Este juego está basado en un «videogame» bastante popular en las salas de juego desde hace algunos años. Nuestra nave espacial sobrevuela un planeta que han invadido los alienígenas; nuestra misión, evidentemente, consiste en destruir a estos invasores.

Al igual que el otro programa de la misma casa que comentamos en esta sección, en éste, las instrucciones y el programa están traducidos (lo poco que hay que traducir ;porque los videojuegos, normalmente, no suelen tener mucho texto). Para jugar debemos tener conectado el *joystick* al puerto dos.

Como hemos comentado, nuestra nave, controlada por el mando, sobrevuela el terreno de un planeta. Por los bordes de la pantalla van apareciendo naves enemigas que nos lanzan bombas; éstas tienen la manía de seguirnos vayamos a dónde vayamos, cosa que también sucede con las naves atacantes. Esta tendencia es imposible de evitar y si no las destruimos pronto acabamos rodeados y chocamos, perdiendo una vida. Existen dos sistemas para destruirlas. El primero es el más expeditivo y deber ser usado únicamente en caso de necesidad, ya que sólo podemos emplearlo una vez. Consiste en pulsar la barra del espacio; al hacerlo estalla una gran bomba «ingeniosa» que destruye todo excepto nuestra nave. El otro sistema es el habitual, que consiste en utilizar nuestros cañones e ir disparando a los enemigos uno a uno y, cuando se acerquen para rodearnos, escapar. Con respecto a esto podemos decir que existe el llamado «efecto frenado», que consiste en que cuando nos movemos de un lado a otro de la pantalla (sin intentar desplazarnos fuera del terreno visible) nos movemos más rápidamente que nuestros oponentes. Pero si intentamos ir al terreno situado a uno de los lados de la pantalla (de modo que ésta haga *scroll*) nuestra velocidad decrecerá sensiblemente, de forma que no podremos huir.

El sistema que se debe emplear para salir de esta situación consiste en situarnos en un lado de la pantalla, esperar a que se acerquen, y cuando ya estén a una distancia que podríamos considerar peligrosa para la integridad de nuestra nave, nos desplazamos rápidamente al otro extremo de la pantalla, giramos y empezamos a disparar a las naves que se acercan.

PUNTUACION:
ADICION: 7
PRESENTACION: 6
GRAFICOS: 5
ACCION: 7



MODEM ACUSTICO PORTATIL
¡NOVEDAD!

La comunicación entre ordenadores es sencilla en cualquier idioma gracias al "MODEM PROTEK". El acoplador acústico provee de una unión fiable entre el ordenador y la línea telefónica. No hay cargas de conectado, la unidad es totalmente portátil. La conexión flexible entre las piezas micrófono y auricular asegura una conexión fiable entre la mayoría de los teléfonos.



INTERFACE PARA IMPRESORAS PARALELO

El nombre de la impresora no queda limitado a su COMMODORE 64. Elija cualquier marca de impresora de salida paralelo, de margarita, matricial e incluso impresoras térmicas, etc.

Características:



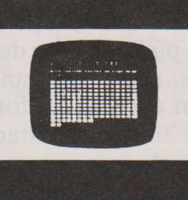
— Compatible con el Software existente.
— El interface se conecta al Port Serie. Dejando libre el port de usuarios.
— Se conecta directamente.
— Compatible Logo COMMODORE.
— Conversión con códigos especiales COMMODORE para impresoras.
— Cambiar número de periférico.
— No necesita alimentación.

ECONO - BUFFER 16 Kbytes PARA IMPRESORAS

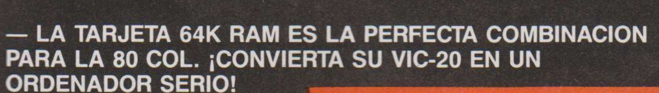
Le permite a su COMMODORE poder seguir utilizándolo sin tener que esperar a que su impresora termine de escribir. Utilizando el disco o cassette.

Características:

— Adaptar a 220/240 VAC 50/60 HZ
— Dimensiones 120×60×80 mm
— Peso 350 gr.
— Longitud cable 45 cm.
— Propio microprocesador.
— Centronics paralelo.
— Copia, numera pág. y pág. sueltas.
— Comunicación 300, 600, 1200, 2400, 3600, 4800, 9600 Baudios



— TARJETA 64 K RAM. VIC 20
— TARJETA 40/80 COLUMNAS VIC 20
— TARJETA 80 COLUMNAS COMMODORE-64
— SLOTS DE EXPANSION (2 Y 5 TARJETAS)
— EXTENSO SOFTWARE 80 COLUMNAS



— LA TARJETA 64K RAM ES LA PERFECTA COMBINACION PARA LA 80 COL. ¡CONVIERTA SU VIC-20 EN UN ORDENADOR SERIO!

PEDIDOS A:

FERRE-MORET S.A.

C/. TUSET, 8 08006 BARCELONA
TEL. (93) 218 02 93

¡SAUCER ATTACK!



Encontrará el juego nº 1 de U.S.A.

en **El Corte Inglés**
y en

En BARCELONA
TELEUNION COMPUTER. C/. Buenos Aires, 57
NOVO-DIGIT. S. A. Aragón, 472
ELECTRONICA JOVALL. Gran de San Andrés, 129
ELECTRO AFICION. Villarroel, 104
ELECTRONICA VIVAS. Mosén Jacinto Verdaguer, 78. Santa Coloma de Gramanet
RADIO DEFOREST. Viladomat, 105
Kios. Aeropuerto de Barcelona
Areas Sants Estación Sants Ferrocarriles
La Librería, S.A. Avda. Sarriá, 2
Librería Artos. Mayor de Sarriá, 2
Leonor Hernández. Reina Elisenda, 2
Librería Bosch. Ronda Universidad, 11
Ediciones Z. Paseo de Gracia, 19
Mallorca. Rambla Cataluña, 86
Libros y revistas BRUGUERA. Rambla Cataluña, 72
XAPS. Balmes, 244
ISLA DEL TESORO. Manuel Girona, 42
MANDRI. Mandri/Cerdanyola
Calvet. Pza. Núñez de Arce
Kiosco Canuda. Ramblas
Librería Catalonia. Rda. S. Pedro fte. Corte Inglés
Kiosco Vanguardia. Paseo de Gracia
Kiosco Drugstore. Paseo de Gracia.
Librería Francesa. Paseo de Gracia
Kiosco LA OCA. Plaza Calvo Sotelo
Kiosco ZANON. Plaza Calvo Sotelo
Kiosco Princesa SOFIA. Frente Hotel Princesa Sofía
Kiosco Intern. Pedralbes. Pº Manuel Girona
Kiosco COCA. Caspe
Kiosco MARTOS. Ramblas
Kiosco Colón. Ramblas
Kiosco Carrillo. Baena Diagonal, fte. Corte Inglés
Kiosco TELE EXPRESS, Pº de Gracia
Kiosco SOLI. Ramblas Plaza Catalunya, 13
Kiosco YA. Plaza Cataluña, 13
Drugstore DAVID. Tuset
Librería Arcadia. Tuset-Pje. Arcadia
Kiosco Corte Inglés. Corte Inglés Diagonal
VIC (BARCELONA). SERVI COMPUT. C/. Moragas, 46 bjs.
VILAFRANCA DEL PENEDES.
RADIO COMPUTER CENTER. Ctra. de Igualada, 21
TARRAGONA. ELECTRONICA RIFE.
C/. Ramón y Cajal, 64
GERONA. REGISCOMPTE, S.A. C/. Emilio Grahit, 17 bis
LERIDA. TELESA. Doctor Fleming, 53
NAVARRA-ANDORRA. Avda. Roncesvalles, 8 PAMPLONA
ZARAGOZA. ADA COMPUTER.
Pº Independencia, 24-26
BILBAO. AYMOSA. Blas de Otero, 45
SESTAO (BILBAO). C/. Vía Galindo, 2
SANTIAGO DE COMPOSTELA. TADEL. C/. Mezonzo, 15
HALT SOFTWARE. C/. Alférez Provisional, 2 ent. E
TORRELAVEGA (CANTABRIA). INFORMATICA SIGLO XXI.
C/. José M. Pereda, 1
BURGOS. CENTRO WELCOME.
Alejandro Rodríguez Valcárcel, 9
MADRID. ABC INFORMATICA. Zurbano, 91 6-B
MADRID. KEY INFORMATICA. Embajadores, 90, tienda
LAS PALMAS DE GRAN CANARIA.
PEDRO GONZALEZ MELIAN. Rosario, 5, Vegueta
PALMA DE MALLORCA. DIMEL, S.A.
Juan de Cremon, 4

Sprites: Los traviesos duendecillos

PRIMERA PARTE

Una de las facilidades más características del Commodore 64 está constituida por los denominados «duendecillos» o *Sprites.* Los *sprites* son formas gráficas que el programador puede definir y controlar de varias maneras, y que proporcionan al C-64 unas amplias capacidades gráficas, por encima del nivel estándar de otros microordenadores.

La mayor parte de las ventajas de los *sprites* derivan del hecho de que son controlados directamente por un elemento del hardware, el chip VIC II. Este chip no es más que una pastilla de silicio sobre la que se han integrado montones de dispositivos electrónicos que, actuando en conjunto, son los responsables de todos los modos gráficos del C-64, incluyendo los *sprites.*

El programador sólo tiene que ocuparse de definir inicialmente la forma y el color del *sprite* y ocuparse luego de ir indicando las posiciones donde quiere colocarlo. Todo el complejo proceso de visualización del *sprite* lo lleva a cabo el chip VIC II, a una velocidad muy superior a la que se podría conseguir haciendo lo mismo desde el BASIC o incluso desde código máquina.

Por si esto fuera poco, el chip VIC II permite controlar de esta forma hasta ocho *sprites* simultáneamente, detecta y señala cuando se ha producido una colisión entre *sprites* o entre un *sprite* y otros elementos de imagen, permite asignar prioridades a los *sprites* de forma que unos se superpongan a otros y ofrece una opción de conseguir *sprites* multicolores. Y todo ello de una forma sencilla mediante la utilización de un conjunto de registros a los que puede acceder el programador desde un programa en BASIC, en código máquina o en cualquier otro lenguaje que permita un acceso directo a la memoria del ordenador.

Vamos a ver paso a paso cómo se manejan los *sprites* con la esperanza de aclarar algunos conceptos que hayan podido quedar oscuros en los no demasiados claros manuales y guías de usuario.

ESTRUCTURA DE UN SPRITE

Si observáramos con una lupa sobre la pantalla del televisor una serie de *sprites,* veríamos que todos ellos están constituidos por un conjunto de pixels (elementos de imagen) que al tomar un color determinado dan forma al *sprite.* Pero, más importante, veríamos que todos ellos sin excepción encajan en un cuadradito del 21 × 24 puntos. Esta es la estructura fundamental de un *sprite,* es decir, un conjunto de puntos que forman un cuadradito de 21 × 24 sobre la pantalla. La persona que define el *sprite* puede hacer que resulte visible cualquiera de los puntos de este cuadradito. Al escoger, para hacerlos visibles, un grupo de puntos determinado, se pueden crear *sprites* de formas y aspectos muy diversos. Los puntos que se decida no hacer visibles, resultarán transparentes y tomarán el color del fondo sobre el que se sitúe el cuadradito.

Ahora que sabemos qué es un *sprite* y cuál es su estructura sobre la pantalla, vamos a ver cómo se representa esta estructura en la memoria del ordenador.

Sabemos que un ordenador maneja la información en forma de valores binarios o bits. Cada bit puede tomar dos valores «0» ó «1» y cada uno de los puntos del cuadradito que constituye el *sprite* puede estar en uno de dos estados, «visible» o «transparente». Parece adecuado entonces que cada punto del *sprite* venga representado por un bit de la memoria.

También sabemos que al ordenador no le gusta manejar los bits uno a uno, sino que prefiere tomarlos en grupos de 8, lo que se llama un *byte.* Si ponemos tres *bytes,* uno a continuación de otro, tenemos una fila de 24 bits. Y si tomamos 21 de estas filas, una debajo de otra, lo que tenemos es un cuadradito de 21 × 24 bits. ¡Exactamente la misma estructura que tenía el *sprite* cuando lo observábamos sobre la pantalla!

Así pues ya hemos descubierto cómo se almacena un *sprite* en la memoria del ordenador. Mediante 21 filas de 3 *bytes,* o lo que es lo mismo, mediante 63 *bytes.*

En realidad hay que añadir otro *byte,* que sirve para indicar el final del *sprite,* y que debe ir con todos sus bits a «0». Con esto un *sprite* queda almacenado en memoria como un conjunto de 64 *bytes.* En la figura 1 hemos representado las dos estructuras, la estructura de puntos sobre la pantalla y su correspondiente estructura de *bytes* en la memoria.

Ya sabemos muchas cosas de los *sprites,* pero vamos a ver alguna más. El C-64 puede manejar hasta 8 *sprites* simultánemante, cada uno de los cuales viene definido por un grupo de 64 *bytes.* ¿Cómo sabe el C-64 dónde está la información de los *sprites?* Bueno, una solución que se le ocurre a cualquiera sería reservar ciertas po-

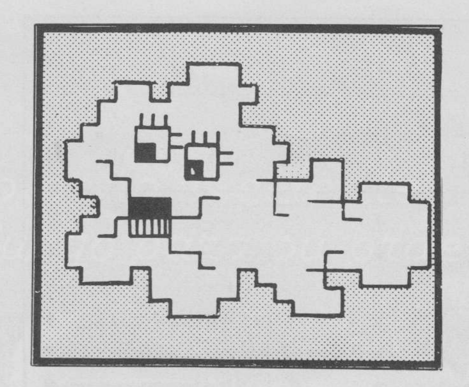
siciones de memoria para los *sprites* y no utilizarlas más que para ellos. Así habría posiciones fijas donde ir a buscar la información.

La solución adoptada, sin embargo, ha sido diferente. Hay 8 nuevos *bytes* que le indican al ordenador dónde comienza la información de cada *sprite*. En estos *bytes* se almacena un número entre 0 y 255. Ese número multiplicado por 64 da la dirección de comienzo de cada *sprite*, relativa al primer *byte* del banco de memoria en el que está trabajando el chip VIC II. Complicado ¿verdad?

Como ejemplo supongamos que el chip VIC II está trabajando en el banco 0. Entonces la dirección de comienzo del banco es precisamente 0. Para buscar la información del *sprite* 3 el ordenador miraría en el puntero del *sprite* 3 (uno de los 8 *bytes* que hemos dicho). Supongamos que este puntero tiene un valor de 150. Entonces el ordenador sabe que el *sprite* 3 comienza en la dirección:

Dirección de comienzo =
150 × 64 + 0 = 9.600

Los punteros de los *sprites* son los ocho últimos *bytes* de los 1K *bytes* de



la memoria de pantalla. Para la pantalla sólo se utilizan 1.000 de los 1.024 *bytes*. Para el que crea que es poco complicado hay que decir que la situación de la memoria de pantalla tampoco es fija y que si cambia la memoria de pantalla también cambian las posiciones de los punteros de los *sprites*.

En la tabla 2 indicamos las posiciones normales de los punteros de los *sprites* (son las que quedan fijadas al encender el ordenador y antes de cambiar nada).

Hasta ahora el programador se tiene que preocupar de definir el *sprite*, de almacenarlo en 64 *bytes* de memoria y de decir dónde comienzan esos 64 *bytes* poniendo un valor en el puntero del *sprite*.

VIDA Y COLOR PARA EL SPRITE

Hemos definido qué es un *sprite* y hemos almacenado la estructura del mismo en la memoria del ordenador. Asimismo hemos puesto un valor en un puntero para indicarle al ordenador donde empieza el *sprite*. Entonces, ¿por qué no aparece todavía en la pantalla?

Bueno, para que aparezca el *sprite* en la pantalla nos quedan tres cosas por hacer. Hay que darle vida y color y hay que indicar en qué punto de la pantalla queremos que aparezca.

Empecemos dándole vida, es decir, activando el *sprite*. Para ello vamos a hablar de un nuevo *byte*, el que lleva la dirección 53269 (D015 en hexadecimal). Este *byte* se llama registro de activación de *sprites* y permite activar o desactivar cada uno de los ocho *sprites*, independientemente unos de otros, ya que cada uno de los ocho bits de los que componen este registro se corresponde con uno de los

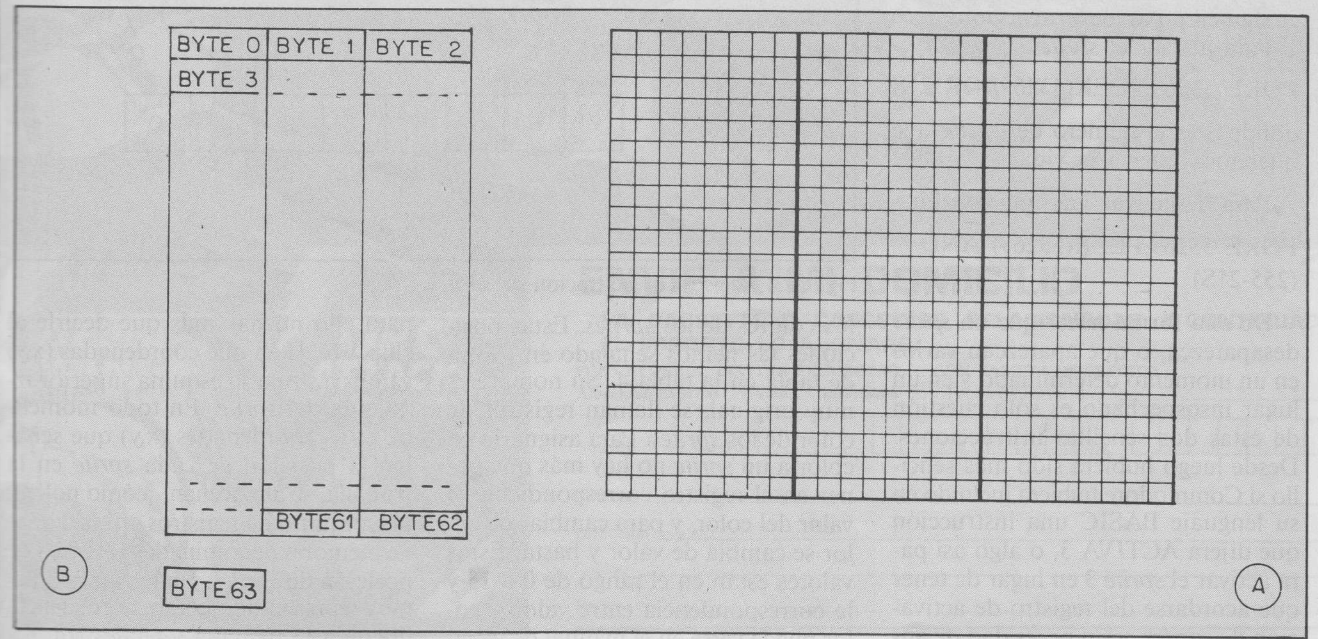


Figura 1. Estructura de un *sprite*, a) como aparece en la pantalla, b) estructura correspondiente de la memoria.

DIRECCIONES DE LOS PUNTEROS DE SPRITES		
SPRITE NUMERO	DIRECCION DECIMAL	DIRECCION HEXADECIMAL
0	02040	07F8
1	02041	07F9
2	02042	07FA
3	02043	07FB
4	02044	07FC
5	02045	07FD
6	02046	07FE
7	0247	07FF

Tabla 2. Direcciones de los punteros de los ocho *sprites.* Estas aparecen al encender el micro, pero pueden cambiarse.

ocho *sprites* (esto puede verse en la figura 3). Activar un *sprite* significa hacerlo visible, permitir que aparezca en la pantalla, y para ello no hay más que poner a «1» el bit correspondiente del registro. Por el contrario, si lo que queremos es desactivar un *sprite,* es decir, hacerlo visible, lo único que tenemos que hacer es poner a «0» el bit correspondiente.

Desde un programa BASIC estas dos operaciones se llevan a cabo con el siguiente par de instrucciones:

Para activar el *sprite*

POKE 53269,PEEK(53269) OR 2↑S

donde S es el número del *sprite* que queremos traer a la vida.

Para desactivar hay que escribir

POKE 53269,PEEK(53269) AND (255-2↑S)

De esta forma hacer que un *sprite* desaparezca, o que aparezcan varios en un momento determinado y en un lugar insospechado es sólo cuestión de estas dos sencillas instrucciones. Desde luego hubiera sido más sencillo si Commodore hubiera incluido en su lenguaje BASIC una instrucción que dijera ACTIVA 3, o algo así para activar el *sprite* 3 en lugar de tener que acordarse del registro de activación de *sprites,* y sin necesidad de andar calculando valores.

¿Y qué hay del color? Bueno, el programador puede darle a cualquiera de los ocho *sprites* cualquiera de los dieciséis colores de que dispone el C-64. El *sprite* 0 puede ser rojo, el 1 amarillo...

Los colores de los *sprites* se almacenanen en *bytes* (qué raro, ¿verdad?). Hay ocho *bytes* o posiciones de memoria fijas en las que se almacenan

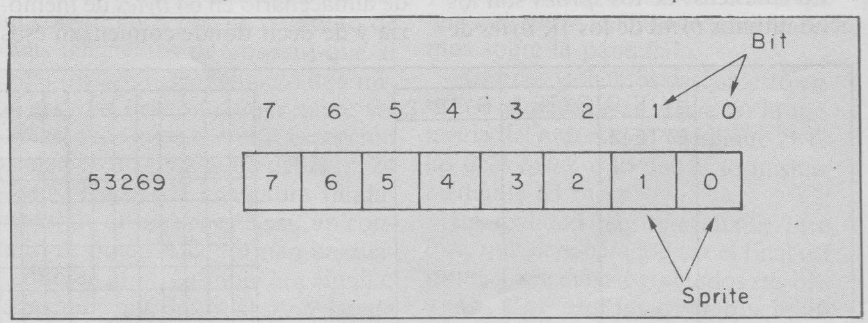


Figura 3. Registro de activación de *sprites.*

los colores de los *sprites.* Estas posiciones las hemos señalado en forma de tabla en la tabla 4. Su nombre es muy original, se llaman registros de color de los *sprites.* Para asignarle un color a un *sprite* no hay más que poner en el registro correspondiente el valor del color, y para cambiar de color se cambia de valor y basta. Estos valores están en el rango de 0 a 15 y la correspondencia entre valor y color puede verse en el manual de usuario que acompaña al C-64.

El proceso de asignar color desde un programa BASIC nos da pie para formular una nueva queja. Hubiera sido sencillo que Commodore hubiera incluido una sentencia que dijera algo así como COLOR 3,7 para indicar que el *sprite* 3 va a ser del color 7 (amarillo). Pues no. Hay que utilizar un POKE y acordarse o buscar la dirección del registro correspondiente al *sprite* 3 para escribir la sentencia correcta, que es:

POKE 53290,7

En el fondo la diferencia no es muy grande, pero resultaría mucho más sencillo de acordarse a la hora de teclearlo y haría que los listados de programas fueran mucho más comprensibles y agradables de leer.

COLOCANDO SPRITES EN LA PANTALLA

El último punto que nos queda por ver antes de pasar a presentar un programita que hace uso de dos *sprites,* es cómo situar un *sprite* en un punto determinado de la pantalla. Bueno, para ello no hay más que decirle al chip VIC II en qué coordenadas (x,y) vamos a situar la esquina superior izquierda del *sprite.* En todo momento, estas coordenadas (x,y) que señalan la posición de cada *sprite* en la pantalla, se almacenan, ¡cómo no!, en un conjunto de registros o posiciones de memoria denominadas registros de posición de *sprites.* En la figura 5 hemos representado cómo se encuentra dividida la pantalla y cuáles son los valores que pueden tomar las coorde-



Exclusivo para lectores de

CLUB DE SOFTWARE

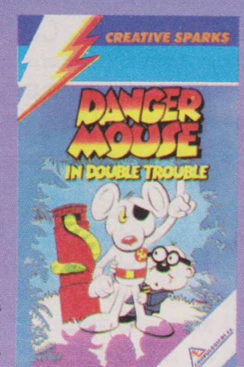
Ha seleccionado ocho excelentes programas de juego para su ordenador COMMODORE 64



Precio: 1.550 ptas.

BLACK HAWK

Usted es un piloto de pruebas altamente entrenado para pilotar el avion mas destructivo hasta ahora creado. Dos pantallas con casi 30 niveles de habilidad



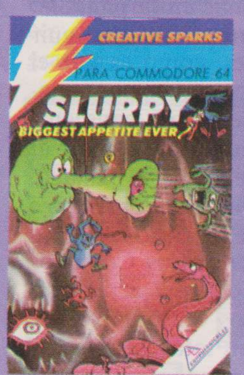
Precio: 1.550 ptas.

DANGER MOUSE

El malvado Baron Silas Greenback ha construido un androide que Danger Mouse debe destruir sin perder un momento.

Guie a Danger Mouse y a Penfold a traves de la espesa jungla hasta el lugar donde se encuentra el Baron, y despues detengale antes de que sea demasiado tarde.

¿Puede usted ayudar a Danger Mouse a salvar el mundo?



Precio: 1.550 ptas.

SLURPY

Slurpy es un adorable y pequeño personaje con un insaciable apetito, que pasa su vida devorando los sabrosos piscolabis que vuelan en sus cavernas.

Ayude a Slurpy a cesar los dolores de hambre y a evitar a sus enemigos en este fantástico juego en código máquina.

- Divertido sonido y gráficos.
- Control joystick.
- Juego para uno o dos jugadores.



Precio: 1.550 ptas.

SPECIAL DELIVERY

Santa Claus se encuentra metido en un terrible lio: se durmio en los laureles, y ahora no tiene suficientes regalos para visitar a todos los impacientes niños.

Ayude a Santa Claus a recoger regalos adicionales y despues a descender sobre los tejados para entregar los regalos introduciéndose por las diversas chimeneas.

- Tres pantallas rodantes.
- Amplia animación gráfica y música.





Precio: 1.550 ptas.

JAVA JIM...

Guie a Java Jim sobre la pantalla excavando perfectos agujeros cuadrados para buscar reliquias antiquisimas.

Usted tendra que evitar los chorros de lava candente que caen continuamente sobre Jim del volcán que se encuentra en activo cerca de el.

- Dos gráficos completos en todas las pantallas de acción.
- 20 niveles de juego.
- Posibilidad de conseguir alta puntuación.
- Juego para uno o dos jugadores.



Precio: 1.550 ptas.

BIRD MOTHER

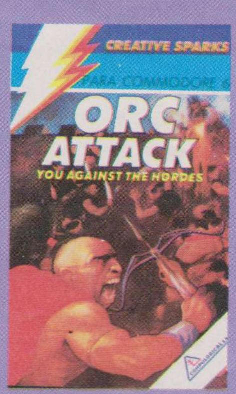
Este es un juego delicadamente original, que toma a la Naturaleza como base para un juego altamente compulsivo y emocionante. El jugador debe guiar a Mamá Pájaro para recoger material para construir su nido y buscar alimento para los polluelos, debiendo protegerles de un terrible halcón depredador.

- Rápida carga en cassette.

Precio: 1.550 ptas.

DELTA WING

Delta Wing es un avanzado simulacro de combate en vuelo. Ha sido diseñado para dar la excelente "sensacion" de los giros y volteretas de la alta velocidad en vuelo. Con 15 controles independientes y aviones de ataque enemigos para derribar, este programa le lleva a usted y a su Spectrum al limite. Delta Wing tambien permite conectar dos ordenadores Spectrum para jugar uno contra otro mediante el interface



Precio: 1.550 ptas.

ORC ATTACK

Defienda su castillo contra las terribles hordas enemigas de los Orcs.

Este juego, 100 por 100 codigo maquina, proporciona un magnifico sonido y graficos que se desarrollan en cuatro pantallas de excitante accion.

- Juego de uno o dos jugadores.
- Posibilidad de conseguir alta puntuación.
- Opcion de teclado o joystick.

ENVIE A MI DOMICILIO

LA CASSETTE O CASSETTES RELACIONADAS A CONTINUACION
EL IMPORTE LO ABONARE:

POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON TARJETA DE CREDITO ☐
American Express ☐ Visa ☐ Interbank ☐ Fecha de caducidad ____________

Número de mi Tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Nombre ____________________________

Dirección ____________________________

Ciudad ____________________________ D.P. ________

Provincia ____________________________

cantidad	producto	ptas.	total

nadas x e y. La x puede tomar cualquier valor entre 0 y 511, mientras que la coordenada y puede variar entre 0 y 255. Al observar la figura nos damos cuenta de que hay ciertos valores de las coordenadas que caen fuera de la pantalla. Esto está pensado así para que sea posible situar un *sprite* de modo que sólo una parte del mismo resulte visible en la pantalla. De este modo se pueden conseguir efectos interesantes, como es que el *sprite* aparezca poco a poco por un lateral, lo que no se podría conseguir si todas las posiciones para situar al *sprite* estuvieran dentro de la pantalla.

Hemos dicho que las coordenadas x,y de cada uno de los ocho *sprites* se guardan en ciertos registros. Como los ocho *sprites* pueden estar activos simultáneamente nos van a hacer falta, en principio, un total de dieciséis registros, ocho para coordenadas x y otros ocho para coordenadas y. Lo que ocurre es que un registro, que es un *byte* de memoria, no puede contener números mayores que 255. Este valor es suficiente para almacenar la coordenada y pero no la x, que puede valer hasta 511. El problema ha sido resuelto con la utilización de otro registro más. En este nuevo registro (de 8 bits como todos los resgistros) se reserva un bit para cada uno de los *sprites*. Con este bit se indica si la coordenada x está por debajo o por encima de 255 según el sistema de numeración binario. Por ello el registro correspondiente se denomina registro de los bits más significativos. Este registro es la posición de memoria 53264, y lo hemos representado en la figura 6.

Para almacenar las coordenadas (x,y) en los registros correspondientes y poder situar el *sprite* en la pantalla, hay que escribir el siguiente conjunto de instrucciones BASIC:

```
H = INT (X/256)   (H=0 si X es menor que 256, si no H=1)
L = X-256*H       (valor que hay que poner en el registro de la X)

POKE 53248 + 2*S,L   (Entrada de la coordenada X)

IF H=1: POKE 53264,PEEK (53264) OR 2↑S
IF H=0: POKE 53264,PEEK (53264) AND (255-2↑S)

POKE 53249 + 2*S,Y   (Entrada de la coordenada Y)
```

Este fárrago de instrucciones es necesario para situar un *sprite*, concretamente el *sprite* N.º S, en las coordenadas x,y de la pantalla. Aunque sea lamentable, no hay más remedio que escribir todas estas instrucciones BASIC, a menos que uno sea un experto y modifique el intérprete de BASIC para que entienda otro tipo de instrucciones.

Ah, se nos olvidaba. Las direcciones de los registros de posición de *sprites*, en los que hay que POKEar las coordenadas, son las que pueden verse en la tabla 7.

REGISTROS DE COLOR DE SPRITES

SPRITE NUMERO	DIRECCIONES DECIMAL	DIRECCIONES HEXADECIMAL
0	53287	D027
1	53288	D028
2	53289	D029
3	53290	D02A
4	53291	D02B
5	53292	D02C
6	53293	D02D
7	53294	D02E

Tabla 4. Direcciones correspondientes a los registros de color.

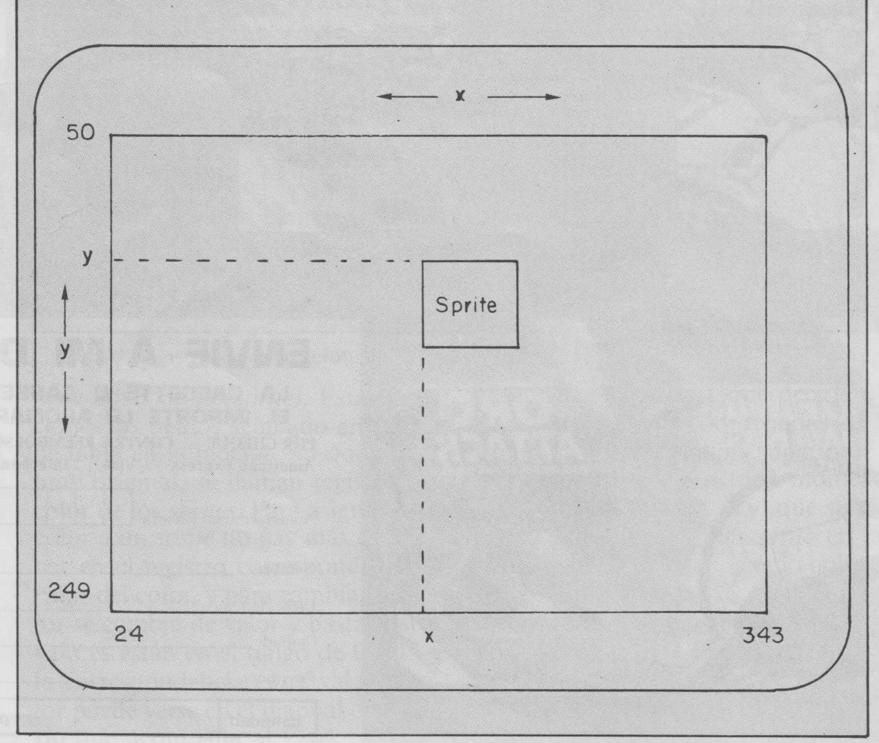


Figura 5. Estructura de la pantalla y coordenadas para los *sprites*.

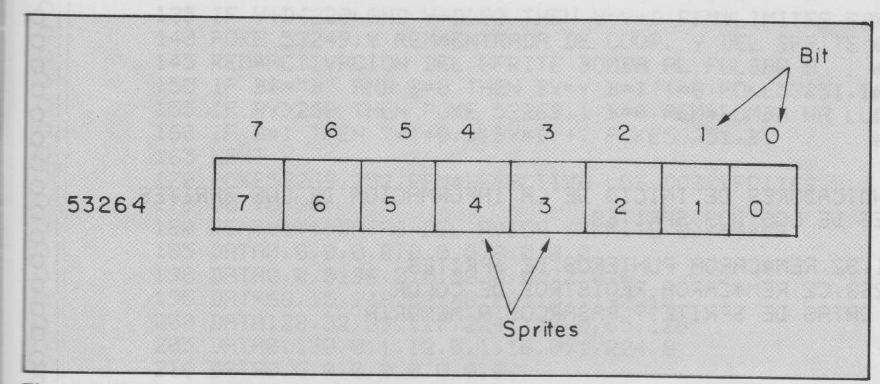
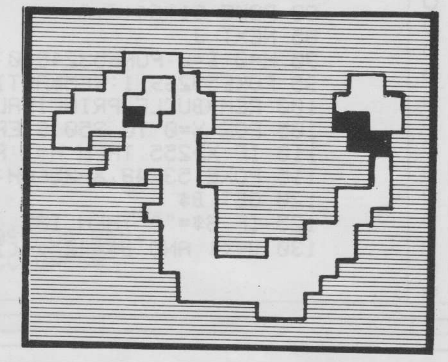


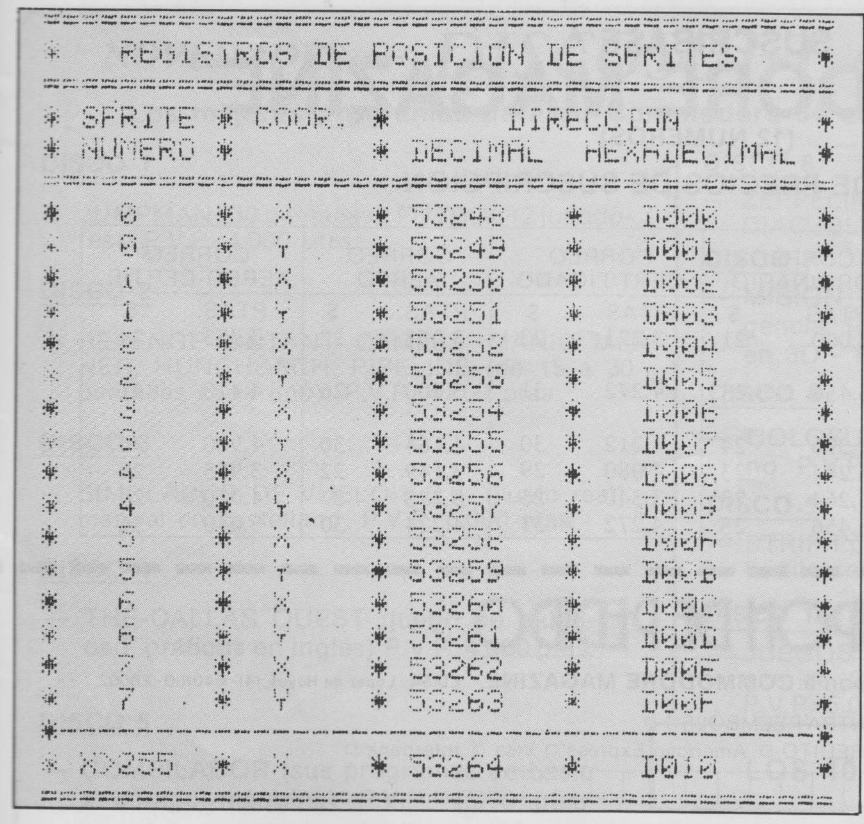
Figura 6. Registro de los bits más significativos de las coordenadas X de cada *sprite.*

UN PROGRAMITA

Como aplicación práctica de todo lo que llevamos dicho, adjuntamos un pequeño programa que hace uso de dos *sprites.* Una serie de REMs hace que sea sencillo seguir su estructura, lo que desde luego recomendamos a aquellos de nuestros lectores que quieran comprender cómo se llevan a cabo todos los pasos de la puesta en funcionamiento de un *sprite.* Además del listado del programa también hemos resumido en un cuadro los distintos pasos de la puesta en marcha de un *sprite,* en el orden en que conviene llevarlos a cabo. Viendo este cuadro uno se da cuenta de que no son tan sencillos los *sprites,* y de que hubiera sido muy buena idea incluir comandos específicos para el manejo de los *sprites* en el lenguaje BASIC.

En una segunda parte de este artículo pasaremos revisión a otra serie de posibilidades de los *sprites,* muy interesantes, que son las que permiten obtener muy buenas prestaciones de estos duendecillos, verdaderos artífices de cualquier buen programa de gráficos.





REGISTROS DE POSICION DE SPRITES

SPRITE NUMERO	COOR.	DIRECCION DECIMAL	DIRECCION HEXADECIMAL
0	X	53248	D000
0	Y	53249	D001
1	X	53250	D002
1	Y	53251	D003
2	X	53252	D004
2	Y	53253	D005
3	X	53254	D006
3	Y	53255	D007
4	X	53256	D008
4	Y	53257	D009
5	X	53258	D00A
5	Y	53259	D00B
6	X	53260	D00C
6	Y	53261	D00D
7	X	53262	D00E
7	Y	53263	D00F
X>255	X	53264	D010

Tabla 7. Direcciones de los registros de posición de *sprites.*

1. Definir el *sprite* y cargarlo en 64 *bytes* de memoria poniendo el último *byte* a cero.
2. Colocar un valor en el puntero de *sprite* correspondiente que le indique al chip VIC II donde comienza el *sprite.*
3. Elegir el color para el *sprite* y almacenar el valor correspondiente en el adecuado registro de color de *sprites* (estos registros son las direcciones de memoria comprendidas entre la 53287 y la 53294).
4. Poner a «1» el bit correspondiente al *sprite* en el registro de activación de *sprites* (dirección de memoria 53269).
5. Definir la posición del *sprite* y almacenarla en los registros de posición de *sprites* (se almacenan las coordenadas x e y).

Tabla 8. Resumen de operaciones para dar vida a un *sprite.*

```
10 REM ***************
15 REM *    SPRITES    *
20 REM *               *
25 REM * DEMOSTRACION *
30 REM ***************
35 REM
40 S1=192:S2=193:REM*INDICADORES DE INICIO DE LA INFORMACION DE LOS SPRITES
45 C1=7:C2=7:REM*COLORES DE LOS DOS SPRITES
50 SP=2:Y=100:PRINT"J"
55 POKE2040,S1:POKE2041,S2:REM*CARGA PUNTEROS DE SPRITES
60 POKE53287,C1:POKE53288,C2:REM*CARGA REGISTROS DE COLOR
65 REM*BUCLE PARA LEER DATAS DE SPRITE Y PASARLOS A MEMORIA
70 FOR J=0 TO 127
75 READ A
80 POKE S1*64+J,A
85 NEXT J
90 H=0:B=0:POKE53248,0:POKE53264,0:REM*COOR. X DEL SPRITE 0
95 POKE53269,1:REM*ACTIVACION DEL SPRITE 0
100 REM*BUCLE PRINCIPAL
105 FOR X=0 TO 350 STEP SP
110 IF X>255 THEN H=1:POKE 53264,PEEK(53264) OR 3
115 POKE 53248,X-256*H:POKE53250,X-256*H:REM*COOR. X DE LOS SPRITES 0 Y 1
120 GET B$
125 IF B$="" THEN 135
130 D=(1 AND B$="Z")-(1 AND B$="X")
```

```
135 IF Y+D<220 AND Y+D>50 THEN Y=Y+D:REM*LIMITES PARA COOR. Y
140 POKE 53249,Y:REM*ENTRADA DE COOR. Y DEL SPRITE 0
145 REM*ACTIVACION DEL SPRITE BOMBA AL PULSAR B
150 IF B$="B" AND B=0 THEN BY=Y:B=1:T=0:POKE53251,BY:POKE53269,3:
155 IF BY>250 THEN POKE 53269,1:B=0:REM*BOMBA HA LLEGADO AL SUELO
160 IF B=1 THEN T=T+0.1:BY=BY+T:POKE53251,BY
165 NEXT X
170 POKE53269,252:REM*DESACTIVA LOS DOS SPRITES
175 GO TO 90
180 REM*****DATAS DEL AVION (SPRITE 0)******
185 DATA0,0,0,0,0,0,0,0,0,0,0
190 DATA0,0,0,96,0,0,80,0,0,72,0,0
195 DATA68,15,248,68,48,20,131,192,15
200 DATA128,32,33,127,224,127,0,65,128
205 DATA0,130,0,1,12,0,1,16,0,1,224,0
210 DATA0,0,0,0,0,0,0,0,0,
215 DATA 0:REM*FIN DEL SPRITE*
220 REM*****DATAS DE LA BOMBA (SPRITE 1)******
225 DATA0,0,0,0,0,0,0,0,0,0,0,0
230 DATA0,0,0,0,0,0,0,0,0,0,0,0
235 DATA0,60,0,0,36,0,0,36,0,0,24,0
240 DATA0,36,0,0,66,0,0,66,0,0,66,0
245 DATA0,66,0,0,66,0,0,66,0,0,126,0
250 DATA0,24,0
255 DATA 0:REM*FIN DEL SPRITE*
260 REM*AL TECLEAR EL PROGRAMA SE PUEDEN QUITAR TODOS LOS REM*
265 REM*YA QUE SOLO SIRVEN PARA EXPLICAR LAS INSTRUCCIONES*
```

Forth para el C-64

LAS ESTRUCTURAS DE CONTROL

Puesto que al describir FORTH decíamos que era un lenguajes estructurado, debemos pasar a analizar brevemente sus estructuras de control. Además de las llamadas a procedimientos (palabras) por nombre que sustituyen con gran ventaja al GOSUB del BASIC, tenemos esencialmente cuatro estructuras de control, alguna de las cuales presenta dos variantes.

La estructura de control condicional viene representada por las palabras *IF, THEN,* y *ELSE* y puede aparecer en dos formas:

IF ... THEN y *IF ... ELSE ... THEN*

En el primer caso, *IF* retira de la pila un número y si es cierto (es decir, distinto de cero) las palabras que se encuentran entre *IF* y *THEN* son ejecutadas; si, por el contrario, el número era nulo, se pasa directamente a ejecutar la palabra que sigue a *THEN.* En la segunda variante sólo uno de los conjuntos de palabras representados por puntos suspensivos es ejecutado: el que se encuentran entre *IF* y *ELSE* si el número retirado por *IF* de la pila era cierto y el que hay entre *ELSE* y *THEN* en caso contrario. Por ejemplo, la palabra:

: *POSITIVO? 0* < IF. "NEGATIVO" ELSE "POSITIVO" THEN ;

usada como n *POSITIVO?* nos dirá si n era un número positivo o negativo. El parecido de los nombres unido a la diferencia sustancial en el uso de estas palabras con respecto al hecho en BASIC puede resultar inicialmente desconcertante para el programador de este último lenguaje. Pero esta estructura responde en forma natural a la propia estructura del FORTH y resulta muy fácil familiarizarse con su correcta utilización.

El equivalente del bucle FOR ... = ... TO ... (STEP ...) NEXT del BASIC viene dado por las estructuras:

DO ... LOOP y *DO ... + LOOP*

La primera diferencia con BASIC consiste en que no hay que proporcionar el nombre del contador del bucle, ya que es siempre I (en realidad la cosa es un poquito más complicada y esta característica no resulta ser una limitación). Veamos cómo funciona todo esto mediante un ejemplo:

: . 1-10 11 1 DO I. LOOP;

La palabra *DO* retira de la pila dos números. El que era el último, el 1 en nuestro caso, será el valor inicial del índice del bucle y el otro, el 11, el valor límite. La palabra I deja sobre la pila el valor corriente del contador de bucle y . lo imprime. Finalmente la palabra *LOOP* incrementa en una unidad el índice y si su valor es menor (!) que el límite transfiere el control a la palabra que sigue a *DO*. Por consiguiente, la palabra *.1-10* producirá la impresión de los número enteros comprendidos entre el 1 y 10, pero no la del 11 (¡otra diferencia con BASIC!). La variante *+ LOOP* toma el incremento positivo, o negativo, de la pila y provoca la vuelta a continuación de *DO* si el nuevo valor del índice es menor, o mayor respectivamente, que el límite. Por ejemplo:

: .10-2/2 0 10 DO 1-2 + LOOP;

es una palabra con el nombre cabalístico de . 10—2/2 que nos quiere indicar que la acción de la misma consiste en la impresión de los números 10 8 6 4 2 por este orden. Nótese la libertad que supone tomar el incremento de la pila, ya que el mismo puede ir variando a lo largo del bucle, ¡algo impensable en BASIC! Otras muchas manipulaciones del índice, e incluso del límite, son posibles, pero en general poco recomendables. También existe la posibilidad de salir prematura pero ordenadamente del bucle por medio de la palabra *LEAVE.*

La estructura *BEGIN ... UNTIL* es un bucle controlado no por un contador sino por un número tomado de la pila por *UNTIL* de forma que si es falso, nulo, se produce un salto hacia atrás hasta la palabra que sigue a *BEGIN,* cuyo único papel consiste en indicar al compilador cuál es esta dirección de salto. Por ejemplo,

: .1-10' 1 BEGIN DUP. 1 + DUP 10 > UNTIL DROP:

es equivalente a *.1-10,* aunque más complicado y sin interés. Pero es fácil imaginar que en otros casos es ésta una estructura muy útil, al igual que su equivalente REPEAT ... UNTIL del PASCAL.

Los bucles *DO ... LOOP y BEGIN ... UNTIL* comprueban la condición de salida al final del bucle que en consecuencia, es necesariamente ejecutado al menos una vez. A menudo, sin embargo, es conveniente realizar tal comprobación antes de iniciar el bucle. En este aspecto FORTH se supera a sí mismo, ya que posee una estructura muy versátil que permite comprobar la condición y forzar eventualmente la salida del bucle en cualquier punto del mismo (sin necesidad de un equivalente del discutido GOTO). Se trata del conjunto formado por las palabras *BEGIN ... WHILE ... REPEAT. WHILE* retira un número de la pila y si es falso provoca un salto a la palabra que sigue a *REPEAT* que, por su cuenta, devuelve siempre el control a continuación del *BEGIN* que ha marcado el inicio del bucle.

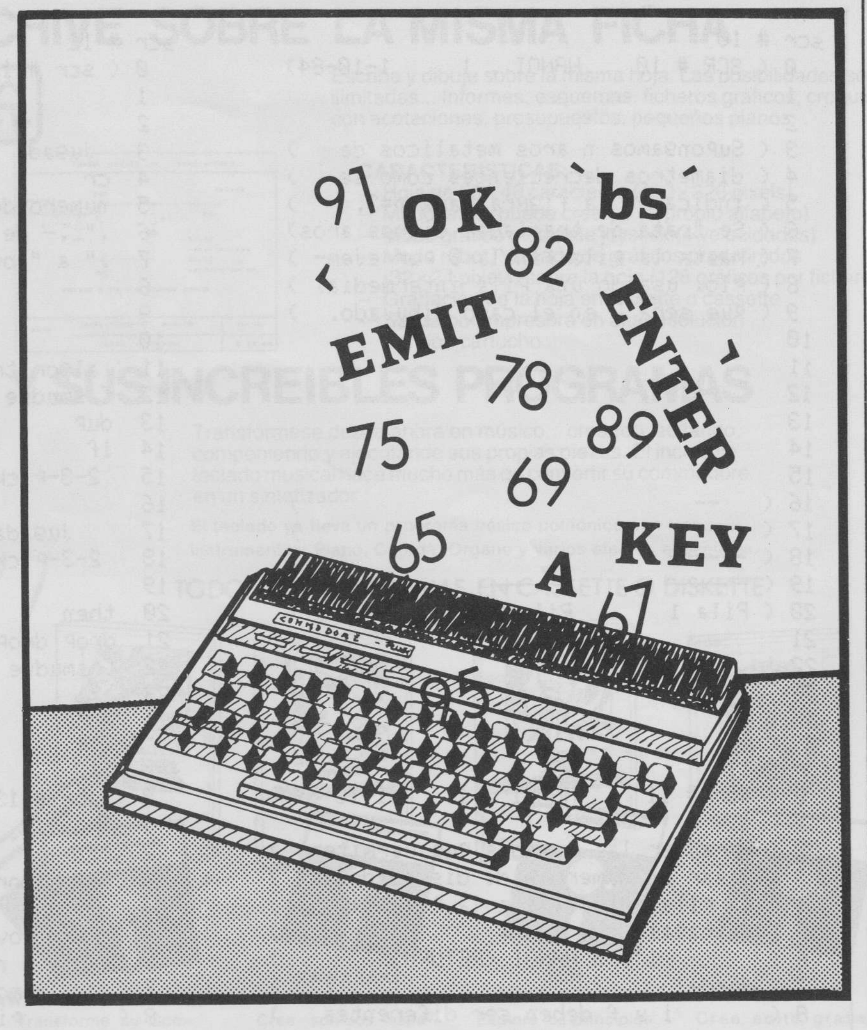
(y 3ª parte)

Por ejemplo, citemos la tercera (y poco interesante) versión de *.1-10.*

: .1—10" 1 BEGIN DUP. DUP 10 WHILE 1 + REPEAT DROP;

A pesar de este ejemplo poco estimulante, es difícil sobrevalorar la flexibilidad de esta estructura que contiene como caso muy particular al WHILE del PASCAL y algunos BASICs evolucionados.

Otras palabras para controlar el desarrollo del programa son, por ejemplo, *EXIT* y *QUIT,* que permiten forzar la salida de una palabra y de todas ellas, respectivamente. Por supuesto que pueden definirse otras palabras de control, pero la extensibilidad de FORTH llega al campo de las propias estructuras de control, pudiendo definirse otras nuevas gracias a la facilidad con la que el proceso de compilación puede ser dirigido. Así es perfectamente posible definir el equivalente de la estructura CASE del PASCAL y (que nos perdonen los fanáticos de la programación estructurada) incluso imitar el denostado GOTO (inexistente e inútil en FORTH), ya que este lenguaje permite casi cualquier cosa, sea o no recomendable.



LA COMPILACION DESDE EL DISCO

Hemos visto hasta ahora cómo podían definirse nuevas palabras; pero ¿qué pasa si la palabra definida resulta ser incorrecta por un error de escritura o, peor aún, de concepto? El espacio de memoria ocupado por la palabra defectuosa, supongámosla llamada *NOMBRE*, puede ser recuperado mediante la frase *FORGET NOMBRE* que también provoca el «olvido» de toda palabra definida con posterioridad a *NOMBRE*. Pero aún queda la tarea, a menudo penosa, de escribir la nueva versión de la palabra o palabras redefinidas que con frecuencia no difieren de la anterior versión sino en algunos puntos. En FORTH, y en general en los lenguajes compilados, resulta por tanto muy útil guardar en disco una copia del llamado «programa fuente», es decir, del programa (conjunto de palabras en FORTH) tal y como se escribe al teclado antes de ser compilado. La posibilidad de compilar desde el disco, y entre otras, la palabra *FORGET* proporcionan un cómodo medio de desarrollo y puesta a punto de programas complejos.

Al objeto de facilitar estas tareas, el FORTH del C-64 dispone de un numeroso grupo de palabras (e incluso de un vocabulario específico: *EDITOR)* que porporcionan, entre otras cosas, un doble editor de textos: uno orientado a la línea y el otro de plena página. Como no tenemos aquí espacio para profundizar en este tema, nos limitaremos a señalar que FORTH divide al disco en tramos de 1.024 octetos llamados «pantallas» («screen» en inglés), ya que su tamaño corresponde (en el C-64 sólo aproximadamente) al de la pantalla de vídeo que permite visualizar simultáneamente el contenido completo de una de tales «pantallas». En ellas puede guardarse el textó fuente de nuestros programas, así como otras muchas cosas que

Forth C-64

```
scr # 10
( SCR # 10     HANOI . 1          1-10-84)
( Supongamos n aros metalicos de    )
( diametros decrecientes como se    )
( indica en la figura para n=3.     )
( Se trata de trasladar dichos aros)
( hasta la pila final, 3 por ejem-  )
( plo, usando una pila intermedia,  )
( que sera 2 en el caso dibujado.   )
(   --                              )
(  ----                             )
( ------                            )
( ______       ______       ______  )
( Pila 1       Pila 2       Pila 3  )
                                  -->
```

```
scr # 11
( scr # 11      HANOI  1      1-10-84 )
( Para ver las jugadas que permiten   )
( pasar un numero n de discos desde   )
( la pila inicial i a la pila final   )
( f, hagase:                          )
(           i f n HANOI <RETURN>      )
( donde:                              )
(        i y f deben ser diferentes   )
(        y comprendidos entre 1 y 3   )
0 variable numero-de-jugada
( Cuenta el numero de jugadas hechas )
: 2-3-pick          ( i,f,n - i,f,n,i,f)
 3 pick 3 pick
;
 intermedio                   ( i,f - m)
 + 6 swap
;      ( Pila intermedia=m=6-i-f)
                                  -->
```

```
scr # 12
( scr # 12  HANOI . 1          1-10-84 )
 jugada                ( i,f,n - i,f,n)
cr
numero-de-jugada dup 1+! ?
." .- de " 3 pick .
." a" over . ." "
;
                       ( Imprime cada jugada)
: algoritmo              ( i,f,n - )
[ smudge ]
dup
if
  2-3-pick over intermedio
  3 pick    1- algoritmo
        jugada
  2-3-pick
        intermedio
  2-3-pick 1- algoritmo
then
drop drop drop
[ smudge ]
;
                                  -->
```

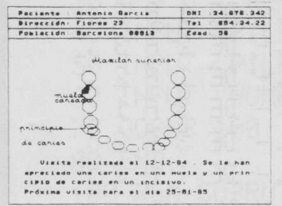
```
scr # 13
( scr # 13     HANOI . 1         1-10-84)
( El algoritmo recurrente usado es:)
( Para mover n piezas de i a f      )
(    Si n no es nulo, entonces      )
(       - mover n-1 piezas de i a la )
(         pila intermedia,          )
(       - mover la pieza de i a f.  )
(       - mover n-1 piezas de la pila)
(         intermedia a f;           )
(    en caso contrario, n=0,        )
(       - no hacer nada.            )
: hanoi                   ( i,f,n - )
 0 numero-de-jugada !
 algoritmo
;
 ( Pone a cero el numero de        )
 ( jugadas y llama al algoritmo)
 ( anteriormente descrito.         )
```

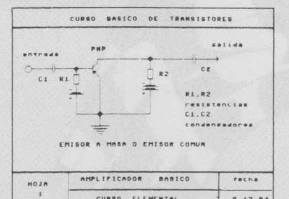
FIGURA 3:
Las «TORRES DE HANOI» en FORTH

Profesionaliza el ocio y disfruta la profesión

DIBUJE Y ARCHIVE SOBRE LA MISMA FICHA

SANDRA





P.V.P. 19.000,-

Escriba y dibuje sobre la misma hoja. Las posibilidades son ilimitadas... Informes, esquemas, ficheros gráficos, croquis con acotaciones, presupuestos, pequeños planos...

CARACTERISTICAS:

- Hoja de 50×40 caracteres (400×320 pixels)
- Modo texto (puede crearse su propio alfabeto)
- Modo gráfico mediante joystick (4 velocidades)
- Modo robot. Volcado de gráficos predefinidos (32×24 pixels) sobre la hoja (128 gráficos por fichero).
- Grabación de la hoja en diskette o cassette
- Salida por impresora en alta resolución
- Versión cartucho

EL TECLADO Y SUS INCREIBLES PROGRAMAS

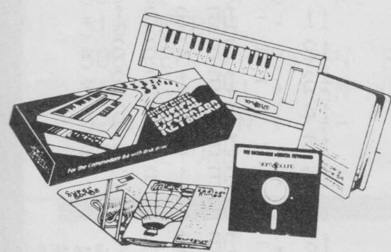


Transfórmese desde ahora en músico... creando, editando, componiendo y ejecutando sus propias piezas. El increíble teclado musical hace mucho más qe convertir su commodore en un sintetizador.

El teclado ya lleva un programa básico polifónico con tres instrumentos: Piano, Cuerda, Organo y varios efectos especiales.

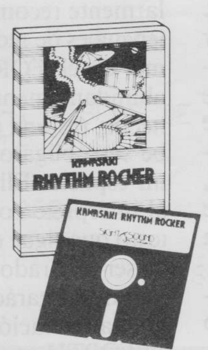
TODOS LOS PROGRAMAS EN CASSETTE O DISKETTE

SIGHT & SOUND Music Software, Inc

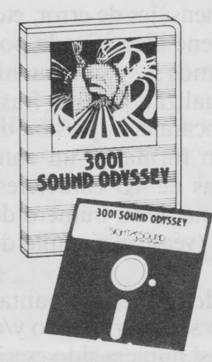




KAWASAKI SYNTHESIZER
Transforme su Commodore-64 en un sintetizador musical programable. Dos diskettes o cassettes.
FANTASTICO



KAWASAKI RHYTHM ROCKER
Cree sonidos espaciales y gráficos multicolores al ritmo de la música electrónica.
EXCITANTE



3001 SOUND ODYSSEY
Explore los principios básicos de la música electrónica sintetizada hasta conseguir crear sus propios efectos de sonido.
EXCEPCIONAL

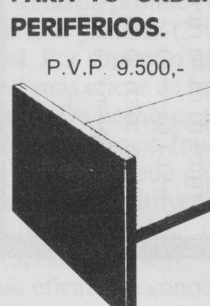


MUSIC PROCESSOR
Cree, edite, grabe, toque y componga sus propios arreglos musicales.
MARAVILLOSO MAESTRO MUSICAL

ARTICULO	VERSION	REFERENCIA	P.V.P.
– INCREIBLE MUSICAL KEYBOARD	DISCO	C64D100	13700
– INCREDIBLE MUSICAL KEYBOARD	CASSETTE	C64T100	12900
– MUSIC PROCESSOR	DISCO	C64D101	9900
– MUSIC PROCESSOR	CASSETTE	C64T101	7900
– KAWASAKI SYNTHESIZER	DISCO	C64D102	12200
– KAWASAKI SYNTHESIZER	CASSETTE	C64T102	9500
– KAWASAKI RHYTHM ROCKER	DISCO	C64D103	10700
– KAWASAKI RHYTHM ROCKER	CASSETTE	C64T103	7900
– 3001 SOUND ODYSSEY	DISCO	C64D104	10500
– 3001 SOUND ODYSSEY	CASSETTE	C64T104	7900

MODULO AUXILIAR DE SOBREMESA PARA TU ORDENADOR Y PERIFERICOS.

P.V.P 9.500,-



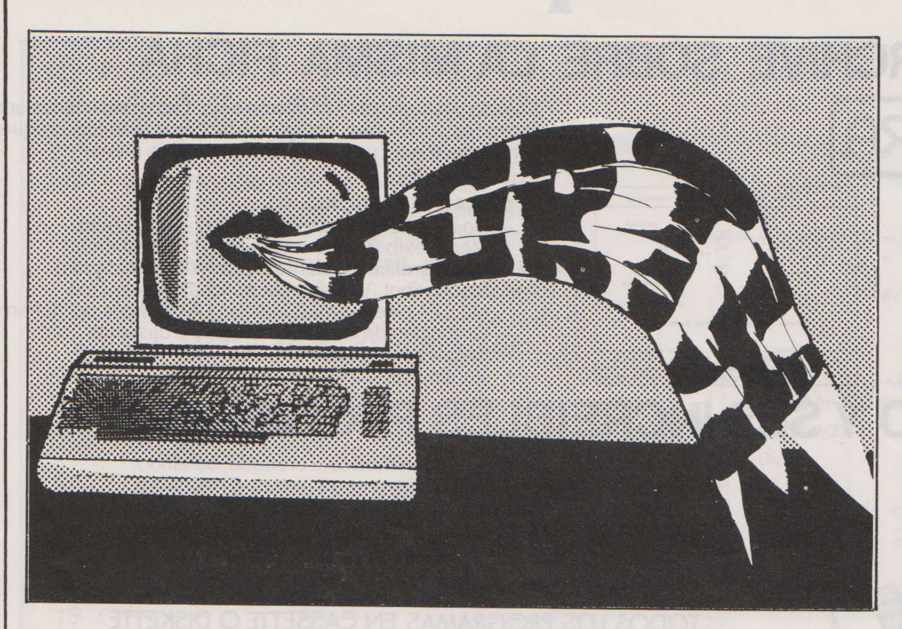


DESEO RECIBIR INFORMACION:
Nombre y dirección:

casa de software s.a.

c/. aragón, 272, 8.°, 6.ª
tel. 215 69 52
08007 barcelona

Forth C-64



incluyen datos, mensajes de error, etc. Cada pantalla viene identificada por un número, digamos n, y su contenido puede ser visualizado con la frase *n LIST* que provocará la impresión de dicha pantalla en forma de un conjunto de 25 líneas de 4Ø caracteres, precedida cada una de su número de orden del Ø al 24 (véase el ejemplo de la figura 3).

El contenido de la n-ésima pantalla puede también ser interpretado y/o compilado como si hubiera sido escrito directamente al teclado, sin más que hacer *n LOAD*. En realidad *LOAD* retira un número de la pila y le dice al intérprete externo que a partir de ese momento considere a la «pantalla» correspondiente a dicho número como si fuera una línea escrita en la pantalla vídeo y que proceda en consecuencia a interpretarla (y eventualmente a compilarla) hasta el final de la misma o hasta que encuentre la palabra - - > que le obliga a continuar interpretando la siguiente «pantalla».

En la figura 3 puede observarse que alrededor del 80% del texto, incluida la mayor parte de las líneas completas, se encuentra entre paréntesis. Esto responde a la costumbre nunca suficientemente encomiada (y particularmente recomendable en el caso de lenguajes de reducida legilibilidad, como es el FORTH) de documentar abundantemente los programas. De hecho, (es una palabra FORTH y debe ser obligatoriamente seguida por un espacio. Ella se encarga de decir al intérprete/compilador que todo el texto que sigue es un comentario y debe ser ignorado hasta la primera aparición del carácter). Nótese también que la ubicación de los comentarios en FORTH es completamente libre y que, en particular, pueden ser intercalados en medio de la definición de una palabra; véase, por ejemplo, la línea 21 de la pantalla 11 en la figura 3.

LAS TORRES DE HANOI

Aunque este ejemplo es bien conocido de todos los lectores, recordemos en qué consiste el problema. Inicialmente se tiene un cierto número de discos apilados por orden de diámetros decrecientes en la pila número 1. El problema consiste en trasladar todos los discos hasta la pila 2 usando como auxiliar únicamente la pila número 3. En cada jugada sólo puede trasladarse un disco y éste únicamente puede ser colocado sobre una pila

```
1 2 3 HANOI

 1 .- DE 1 A 2 .
 2 .- DE 1 A 3 .
 3 .- DE 2 A 3 .
 4 .- DE 1 A 2 .
 5 .- DE 3 A 1 .
 6 .- DE 3 A 2 .
 7 .- DE 1 A 2 .

1 2 5 HANOI

 1 .- DE 1 A 2 .
 2 .- DE 1 A 3 .
 3 .- DE 2 A 3 .
 4 .- DE 1 A 2 .
 5 .- DE 3 A 1 .
 6 .- DE 3 A 2 .
 7 .- DE 1 A 2 .
 8 .- DE 1 A 3 .
 9 .- DE 2 A 3 .
10 .- DE 2 A 1 .
11 .- DE 3 A 1 .
12 .- DE 2 A 3 .
13 .- DE 1 A 2 .
14 .- DE 1 A 3 .
15 .- DE 2 A 3 .
16 .- DE 1 A 2 .
17 .- DE 3 A 1 .
18 .- DE 3 A 2 .
19 .- DE 1 A 2 .
20 .- DE 3 A 1 .
21 .- DE 2 A 3 .
22 .- DE 2 A 1 .
23 .- DE 3 A 1 .
24 .- DE 3 A 2 .
25 .- DE 1 A 2 .
26 .- DE 1 A 3 .
27 .- DE 2 A 3 .
28 .- DE 1 A 2 .
29 .- DE 3 A 1 .
30 .- DE 3 A 2 .
31 .- DE 1 A 2 .
```

FIGURA 4:
Las «TORRES DE HANOI» para n = 3,5

```
5 REM *** HANOI . 1 ***
10 DIM D(100) : REM PILA
20 SP = 3 : REM PUNTERO DE PILA
30 N = 0 : REM NUMERO DE JUGADA
40 INPUT "PILA INICIAL (1-3) "; D(1)
50 INPUT "PILA FINAL (1-3) "; D(2)
60 INPUT "NUMERO DE DISCOS "; D(3)
70 GOSUB 100
80 END
100 REM SUBRUTINA PRINCIPAL
110 IF D(SP) = 0 THEN 300 : CASO TRIVIAL
120 D(SP+1) = D(SP-2)
130 D(SP+2) = 6-D(SP-2)-D(SP-1)
140 SP = SP+3
150 D(SP) = D(SP-3)-1
160 GOSUB 100 : REM AUTOLLAMADA PARA N-1 DISCOS DE INICIAL A INTERMEDIO
170 GOSUB 500 : REM JUGADA
180 D(SP+1) = 6-D(SP-2)-D(SP-1)
190 D(SP+2) = D(SP-1)
200 SP = SP+3
210 D(SP) = D(SP-3)-1
220 GOSUB 100 : REM AUTOLLAMADA PARA N-1 DISCOS DE INTERMEDIO A FINAL
300 SP = SP-3 : REM ACTUALIZA PUNTERO DE PILA
310 RETURN
500 REM IMPRESION  EN PANTALLA DE LA JUGADA
510 N = N+1 : REM NUMERO DE JUGADA
520 PRINT N ; ".- " ; "DE " ; D(SP-2) ; " A " ; D(SP-1) ; "."
530 RETURN
```

FIGURA 5:
Las «TORRES DE HANOI» en BASIC

vacía u otro disco de diámetro superior.

Como es bien conocido el algoritmo recursivo de la «pantalla» 13 en la figura 3 proporciona siempre la única solución posible en un número mínimo de jugadas (2^n — 1) para n discos). Por supuesto que no es el único ni el más eficaz de los algoritmos posibles. De hecho es fácilmente mejorable en eficacia (por ejemplo, es fácil ver que puede ganarse tiempo de ejecución sustituyendo el caso trivial sin llamada recursiva n = 0 por el correspondiente a n = 1) y la versión más eficaz que conocemos no es ni siquiera recursiva. Pero toda ganancia en eficacia (entendida un tanto parcialmente como rapidez de ejecución) redunda necesariamente en detrimento de la claridad del algoritmo, cuya versión más elegante y sencilla es la aquí utilizada.

La figura 3 presenta una posible realización práctica de este algoritmo en FORTH. Como está abundantemente documentada y todas las palabras que allí aparecen (con la excepción que discutiremos a continuación) ya han sido comentadas, esperamos que el lector pueda entenderla.

Las palabras no explicadas hasta ahora son las tres de la frase *[SMUDGE]* que aparece al principio y al final de la definición de *ALGORITMO*. Su utilización es imprescindible para permitir que *ALGORITMO* pueda ser usado, por dos veces, en su propia definición, que por tanto es recursiva. De no haberse incluido esta frase, al llegar el compilador a la palabra *ALGORITMO* de la línea 16 de la «pantalla» 12, la hubiera considerado como inexistente en el diccionario (ya que su definición es aún incompleta) y hubiera forzado un error. Esta característica de FORTH es útil cuando se quiere redefinir parcialmente una palabra anterior (perteneciente por ejemplo al diccionario de base) usando para ello la versión an-

Forth C-64

tigua de la misma, pero obliga a utilizar la ya mencionada frase en las definiciones recursivas.

Las soluciones del problema para n = 3 y n = 5 se presentan en la fihura 4, mientras que las figuras 5 y 6 presentan realizaciones del mismo algoritmo en BASIC y LOGO respectivamente. A la palabra FORTH *ALGORITMO* le sorresponde la subrutina BASIC de las líneas 100—310 y el procedimiento LOGO MOVER. Análogamente, el equivalente BASIC de la palabra FORTH *JUGADA* viene dado por las líneas 500—530 y corresponde al procedimeinto MUEVE del LOGO.

Para n = 10 los tiempos de ejecución han sido los siguientes:

—FORTH 1 min. 51 sg. —BASIC: 3 min. 17 sg.

—LOGO: 3 min. 22 sg.

Puesto que el programa debe imprimir muchos números (3 x 1023 para se más precisos) y ya habíamos señalado que FORTH es más lento que BASIC en este menester, hemos creído interesante averiguar qué parte del tiempo de ejecución corresponde a la impresión y cuál al resto del algoritmo. Para ello hemos cronometrado los tiempos de ejecución tras heber suprimido las líneas 17 de la «pantalla» 12 en FORTH, 170 en BASIC y la que contiene MUEVE: IF del procedimiento MOVER del LOGO. Los resultados son bastante interesantes:

—FORTH: 0 min. 06 sg. (sí, ¡sólo seis segundos!).

—BASIC: 1 min. 47 sg.

—LOGO: 1 min. 01 sg.

Con ello comprobamos que la impresión de números es algo más lenta en FORTH (¡pero mucho más el LOGO!) y que en la realización de algoritmos recursivos FORTH se encuentra realmente a sus anchas, a pesar de la pequeña complicación que presenta la compilación de los mismos.

También puede resultar interesante comparar cómo se ha abordado este algoritmo recursivo en los tres lenguajes considerados. En este tipo de programas el problema clave radica en el

```
TO HANOI :I :F :N
 MAKE "C 0
 MOVER :I :F :N
END

TO MUEVE :I :F
 MAKE "C :C + 1
 PRINT ( SE :C [.- DE] :I [A] :F [.] )
END

TO MOVER :I :F :N
 IF :N = 0 STOP
 MOVER :I 6 - :I - :F :N - 1
 MUEVE :I :F
 MOVER 6 - :I - :F :F :N - 1
END
```

FIGURA 6:
Las «TORRES DE HANOI» en LOGO

tratamiento de los parámetros y variables correspondientes a la palabra, subrutina o procedimiento recursivo en cada una de sus diversas «encarnaciones». Este problema es resuelto en LOGO y en PASCAL por medio de las variables locales, mientras que FORTH hace uso, en forma particularmente eficaz como acabamos de ver, de la pila.

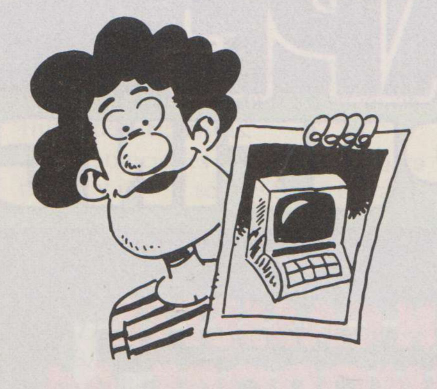
Por su lado, BASIC no ofrece ninguna facilidad para utilizar variables locales (excepto en el caso de algunos BASICs evolucionados y con la sentencia DEFFN), por lo que hemos debido de recurrir a la emulación de una pila mediante la variable unidimensional D y el puntero SP. En honor a la verdad, hemos de reconocer que en esta simulación hemos seguido con demasiada proximidad a FORTH. Esto tiene la ventaja de ayudar a comprender cómo trabaja dicha versión del algoritmo, pero resulta algo injusto para el BASIC, dentro del cual hubiera resultado más eficaz definir tres pilas controladas por el mismo puntero (y agrupar varias sentencias por línea, ...). Aún más, mientras que en FORTH la transformación del algoritmo recursivo en una versión no recursiva difícilmente permitirá arañar algo de tiempo a los menos de 6 segundos dedicados en el ejemplo considerado a la tarea de encadenar palabras, la misma transformación realizada en BASIC hubiera permitido recuperar una parte interesante del minuto y tres cuartos dedicados por este lenguaje a la misma función.

BIBLIOGRAFIA SUCINTA

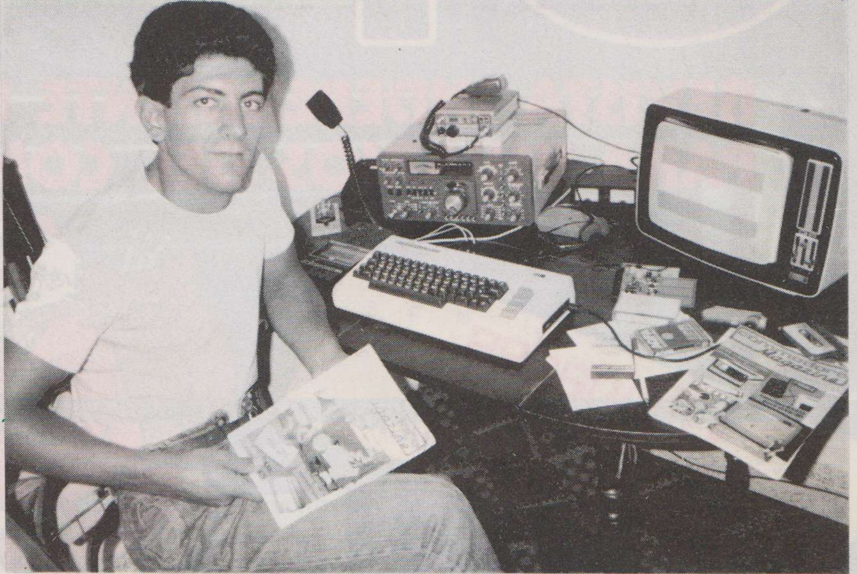
— R. G. Loeliger, «Threaded interpretive languages», Byte Books, 1981.
— L. Brody, «Starting FORTH», Prentice-Hall, 1981.
— P. Chirlian, «Démarrer en FORTH», P.S.I. 1984.
— K. Knecht, «Introduction to FORTH», Howard W. Sams, 1983.
— A. Pinaud, «Programmer en FORTH», P.S.I. 1983.
— O. Bishop, «Exploring FORTH», Granada, 1984.
— W.P. Salman, O. Tisserand et B. Toulout, «FORTH», Eyrolles, 1983.
— P. Courtois, «Le concept FORTH», Editests, P.S.I. 1983.

por Juan María y José Miguel AGUIRREGABIRIA.
Departamento de Física de la Facultad de Ciencias y Escuela Universitaria del Profesorado de EGB, Universidad del País Vasco.

Envianos la foto de tu ordenador



Nuestros commodoreros siguen expresimiéndose el cerebro para conseguir ese rincón ideal donde colocar su colección de cables, pantallas, teclados y demás trastos. Cada uno se las ingenia como puede. Y tú, ¿cómo has montado tu «refugio commodorero»? Esperamos vuestras cartas contándonoslo.



ALBOLOTE (GRANADA)

¡Estimados amigos!

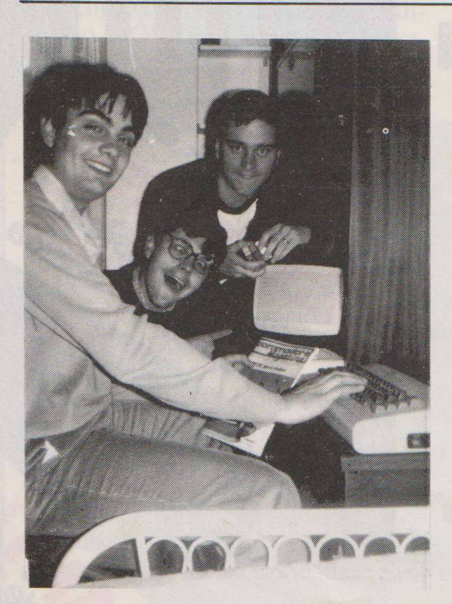
Me gustaría que mi fotografía saliera en la revista de Febrero, y si me la pudiérais mandar sería estupendo. Ahora os contaré de lo que va en mi mesa de operaciones:

La misma consta del Ordenador Commodore VIC-20 estándar, el cual lo empleo para toda clase de programas, está conectado a una televisión marca ELBE de 20 pulgadas en blanco y negro. El *cassette* que uso es un Sanyo estándar con la adaptación que publicásteis en el número uno. Delante de la pantalla se ve un circuito impreso de fabricación casera para la recepción y transmisión de telegrafía CW y RTTY que funciona con un programa efectuado para tal tema. A la izquierda un equipo transmisor-receptor marca Yaesu-FT-901-DM de bandas bajas o decamétricas, como queráis llamarlo, y arriba un Estándar-C-58 de 144 MHZ. Qué más puedo deciros, que soy radioaficionado, EA-2-BKK es mi indicativo oficial.

Ahora los halagos merecidos: que hacéis una revista demasié e instructiva.

Si podéis devolverme la foto me ahorro 60 calas y si no me aguanto. Un fuerte saludo de vuestro amigo.

Paco



MANRESA

Apreciados amigos:

Soy Carlos Basullas de Manresa y como podéis observar os envío la foto de casi toda la familia commodorana de mi casa.

Yo la he titulado El listo, el tonto y el feo.

Aunque no se observa muy bien, porque el televisor es en blanco y negro está en pantalla el programa ideado en mi azotea «Clasifútbol» que ya deben haber recibido para concursar.

El ordenador está montado en mi habitación al lado de la cama, por si te entra sueño pasando programas y en una estantería donde también se halla el T.V., la unidad de *casette* y el transformador. Todos los cables pasan por detrás y convergen en un estabilizador del que tomo la corriente.

Esta le llega por un cable claveteado a la pared (para evitar la gravedad selectiva de J. Raf Martín) y contacto directamente a una toma de corriente de la pared (o sea un enchufe).

El de las gafas es mi hermano, y el del cubo de rubik mi cuñado; ya ves, todos aficionados al mundo informático.

Mi hermano disfruta mucho dándole a la tecla on/off del ordenador cuando estoy pasando algún programa, pero para más goce espera a que esté terminado.

Atentamente
Carlos Basullas Martínez

No se caliente la "CABEZA"

SEIKOSHA

IMPRESORAS

ESTOS SON NUESTROS MODELOS:

Modelo	Velocidad	Columnas	Tipos de letra	Interface	P.V.P.
GP-50	40 cps	46	2	A-Paralelo AS-Serial S-Spectrum	A-25.900 AS-29.900 S-28.900
GP-500	50 cps	80	2	A-Paralelo AS-Serial	A-47.900 AS-49.900
GP-550	86 cps	80-136	18	A-Paralelo	A-59.900
GP-700	50 cps	80-106	3	A-Paralelo	A-89.900
BP-5200	200 cps	136-272	18	Paralelo y serial	199.000
BP-5420	420 cps	136-272	18	Paraleo y serial I-IBM PC	299.000 I-299.000

Disponemos de interfaces opcionales para todos los modelos: IBM PC, COMMODORE 64, ZX SPECTRUM, ATARI, DRAGON 64, SHRAP MZ 700, SPECTRAVIDEO, NEW BRAIN, APPLE, ETC...

Nuestra calidad es "SEIKO"; nuestros precios, únicos. Si desea más información, consulte con nuestro distribuidor más cercano, o llame o escriba a:



Dirección comercial:
Av. Blasco Ibáñez, 114-116.
46022-Valencia.
Tel. (96) 372 88 89.
Télex 62220

Delegación en Cataluña:
C/ Muntaner, 60, 4, 1.
08011-Barcelona.
Tel. (93) 323 32 19.